Commonsense Reasoning about Containers using Radically Incomplete Information

Ernest Davis DAVISE@CS.NYU.EDU
Computer Science Dept., New York University, 251 Mercer St. New York, NY 10012 USA

Gary Marcus GARY.MARCUS@NYU.EDU
Psychology Dept., New York University, New York, NY 10012 USA

Noah Frazier-Logue N.FRAZIER.LOGUE@NYU.EDU
College of Arts and Science, New York University, New York, NY 10012 USA

Abstract

In physical reasoning, humans are often able to carry out useful reasoning based on radically incomplete information. One physical domain that is ubiquitous both in everyday interactions and in many kinds of scientific applications, where reasoning from incomplete information is very common, is the interaction of containers and their contents. We have developed a preliminary knowledge base for qualitative reasoning about containers, expressed in a sorted first-order language of time, geometry, objects, histories, and actions. We have demonstrated that the knowledge suffices to justify a number of commonsense physical inferences, based on very incomplete knowledge.

1. Physical Reasoning Based on Radically Incomplete Information

In physical reasoning, humans, unlike programs for scientific computation, are often able to carry out useful reasoning based on radically incomplete information. If AI systems are to achieve human levels of reasoning, they must likewise have this ability. The challenges of radically incomplete information are often far beyond the scope of existing automated reasoners based on simulation (Davis & Marcus, 2016); rather they require alternative reasoning techniques specifically designed for incomplete information.

As a vivid example, consider the human capacity to reason about containers — boxes, bottles, cups, pails, bags, and so on — and the interactions of containers with their contents. For instance, you can reason that you can carry groceries in a grocery bag and that they will remain in the bag with only very weak specifications of the shape and material of the groceries being carried, the shape and material of the bag, and the trajectory of motion. Containers are ubiquitous in everyday life, and children start to learn how containers work at a very early age (Hespos & Baillargeon, 2001) (figure 1).[1]

[1] Ironically, the working of a baby bottle nipple is beyond the scope of this paper.

Containers likewise are central in a wide range of applications and domains.[2] For example, in a separate study we have recently begun of the reasoning needed to understand a biology textbook (Reece, et al., 2011), we find that physical containers of many different kinds and scales appear in domains relevant to biology. Some examples:

- The membrane of a cell is a container that holds the contents of the cell. Many of the primary processes in the cell are concerned with bringing material into the container and expelling material from the container.
- The skin or other outer layer of an animal is a container for the animal. Again, many of the central life processes — eating, breathing, excreting — deal with transporting material into and out of the container.
- In a discussion of speciation (p. 493), it is mentioned that a subpopulations of a water creature can be isolated if the water level of a lake falls, dividing it into two lakes. Here the container is the lake bed, and the phenomenon depends on the somewhat non-obvious fact that a liquid container that bounds a single connected region at one level may bound two regions at a lower level (figure 2).

Figure 1: Infant learning about containers

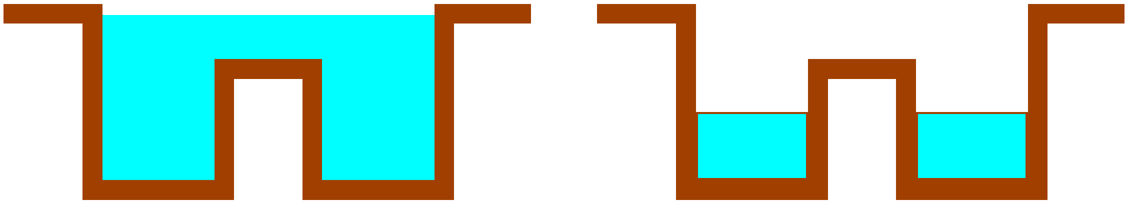


Figure 2: A lake divides into two lakes when the water level falls

[2] Containment is also often used metaphorically. For instance, Lakoff and Johnson (1980) and Reddy (1979) discuss the use of containment as a metaphor for the relation between a linguistic expression and its meaning; e.g. "Your argument has no content". Similarly, in the context of computers, the relation between a memory location such as a variable and its value is often conceptualized as containment.

In this paper we describe the initial stages of development of a knowledge-based system for reasoning about manipulating containers, in which knowledge of geometry and physics and problem specifications are represented by propositions. Below, we outline the system, and show that this approach suffices to justify a number of commonsense physical inferences, based on very incomplete knowledge of the situation and of the dynamic laws that govern the objects involved. These inferences have been automatically verified using the first-order theorem prover SPASS (Weidenbach, et al., 2009).

1.1 Incomplete information

The issues of complete and incomplete information can easily be misunderstood, so let us make clear what we have in mind. Of course, few representations are truly complete or entirely precise; in virtually any representation, some aspects are omitted, some are simplified, and some are approximated. However, techniques such as simulation, or STRIPS-like representations, require that the initial conditions of the scenario and that the dynamics of the microworld be fully specified *relative to a given level of description*. That is, the representational framework specifies some number of critical relations between entities and properties of entities. A complete representation of a situation relative to that framework enumerates all the entities that are relevant to the situation, and specifies all the relations in the framework that hold between those entities. The description must be detailed and precise enough that the situation at the next time step is likewise fully specified, in the same sense.

For instance, the standard blocks world representation omits the size, shape, and physical characteristics of the blocks involved, and the trajectory of the actions. Situations are describe purely in terms of the predicate On(t,x,y) (object x is on object y at time t) and actions are described in terms of Puton(t,x,y) (the agent puts object x onto y at time t). However, the dynamic theory is a complete account at this level of description; that is, a complete enumeration of the On relations that hold in one situation completely determines what actions are feasible, and determines all the On relations that will hold once the action is executed. Additionally, most projection and most planning problems provide a complete enumeration of the On relations that hold in the initial situation.

By contrast, in the theory that we develop in this paper, both general domain axioms and problem specifications may give full specifications of some of the features involve, but leave others partialy specified or wholly unspecified. For instance, inference 1 (section 8) specifies that initially object Ox1 is inside box Ob1, but it does not specify whether or not there are any other objects inside Ob1 nor does it specify whether Ox1 is inside box Ob1, nor does it specify the spatial relation of the agent to either of these. The physical laws given specify that if the agent drops an object that it is holding, the object will end up in a stable state, but the theory does not in general specify where it will end up, or where it will pass through while it is falling, or how it might impact other objects. The theory does support the inference that if it is inside an open container when dropped, it will remain inside the container, and not come into contact with any object outside the container. Some necessary conditions and some sufficient conditions are given for the feasibility of the agent being able to move from a starting to an ending positions are given, but the necessary conditions are much weaker than the sufficient conditions; in many cases, it is indeterminate.

2. Containers

We begin with a general discussion of the properties of containers as encountered in everyday situations and of the characteristics of commonsense reasoning about containers.

A container can be made of a wide range of materials, such as rigid materials, paper, cloth, animal body parts, or combinations of these. The only requirement is that the material should maintain its shape to a sufficient degree that holes do not open up through which the contents can escape. Under some circumstances, there can even be a container whose bottom boundary is a *liquid*; for instance, an insect can be trapped in a region formed by the water in a basin and an upside-down cup. A container can also have a wide range of shapes (precise geometric conditions for different kinds of containers are given in section 6.1.)

The material of the contents of a container is even less constrained. In the case of a closed container, the only constraint is that the material of the contents cannot penetrate or be absorbed into the material of the container (e.g. you cannot carry water in a paper bag or carry light in a cardboard box); and that the contents cannot destroy the material of the container (you cannot keep a gorilla in a balsa wood cage). Using an open container requires additionally that the contents cannot fly out the top (Davis, 2011). Using a container with holes requires that the contents cannot fit or squeeze through the holes.

Those are all the constraints. In the case of a closed container, the material of the contents can be practically anything with practically any kind of dynamics. For instance, you can infer that an eel will remain inside a closed fish tank without knowing anything at all about the mechanisms that eels use to swim or about the motions that are possible for eels.

A container can serve many different purposes, including: carrying contents that are difficult or impossible to carry directly (e.g. a shopping bag or a bottle); ensuring that the contents remain in a fixed place (e.g. a crib or a cage); protecting the contents against other objects or physical influences (e.g. a briefcase or a thermos bottle); hiding the contents from inspection (e.g. an envelope); or ensuring that objects can only enter or exit through specific portals (e.g. a tea-kettle). In some cases it is necessary that some kinds of material or physical effects can either fit through the portals or pass through the material of the container, while others cannot. For instance, a pet-carrying case has holes to allow air to go in and out; a display case allows light to go in and out but not dust.

There are four primary kinds of physical principles involved in all of these cases. First, matter must move continuously; if the contents could be teleported out of the container, as in Star Trek, these constraints would not apply. Second, the contents (or the externality being kept out, such as dust) cannot pass through the material of the container. Third, there are constraints on the deformations possible to the shapes of the container and of the content. Fourth, in the case of an upright open container, gravity prevents the contents from escaping.

Simple, natural examples of commonsense physical reasoning reveal a number of important characteristics (Davis & Marcus, 2014).

First, human reasoners can use very partial spatial information. For example, consider the text, "There was a beetle crawling on the inside of the cup. Wendy trapped it by putting her hand over the top of the cup, then carried the cup outside, and dumped the beetle out onto the lawn." A reader understands that the cup and the hand formed a closed container for the beetle, and that Wendy removed her hand from the top of the cup before dumping the beetle. Thus, qualitative spatial knowledge about cups, hands, and beetles suffices for interpreting the text; the reader does not require the geometry of these to be specified precisely.

Second, human reasoners can often infer that a material is confined within a closed container even if they have only a vague idea of the physics of the material of the container and almost no idea at all of the material of the contents. For example, the text above can be understood by a reader who does not know whether a “beetle” is an insect, a worm, or a small jellyfish.

Third, human reasoners can predict qualitative behavior of a system and ignore the irrelevant complex details; unlike much software, they are often very good at seeing the forest and not being distracted by the trees. For example, if you pour water into a cup, you can predict that, within a few seconds it will be sitting quietly at the bottom of the cup; and you do not need to trace through the complex trajectory that the water goes through in getting to that equilibrium state.

Finally, knowledge about containers, like most high-level knowledge, can be used for a wide variety of tasks in a number of different modalities, including prediction, planning, manipulation, design, textual or visual interpretation, and explanation.

The theory developed in this paper shares these properties, though certainly with much less range and flexibility than a human reasoner. By contrast simulation models almost always require precise physical and spatial information; generate highly detailed, precise predictions; and are aimed almost exclusively at the task of projection. (The limits of simulation models are discussed further in section 5.)

Section 3 of this paper will discuss the overall architecture and goals of our theory of physical reasoning. Section 4 discusses the prospects for using this theory in an implemented automated reasoner. Section 5 explains the advantages of the theory presented here over a theory based on simulation. Section 6 will give a preformal sketch of the physical microworld. Section 7 comprises this majority of this paper; it is a detailed axiomatization of our theory. Section 8 presents five sample inferences and sketches the proofs of the inferences from the theory in section 7 and the validation of the proofs using the SPASS theorem prover. Details of the proofs and the validation are given in an online supplement. Section 9 discusses what is involved in establishing the consistency of this theory. Section 10 discusses related work. Section 11 reviews our conclusions and sketches the major issues for future work.

3. Physical reasoning: Overall architecture.

We conjecture that, in humans, physical reasoning comprises several different modes of reasoning, and we argue that machine reasoning will be most effective if it follows suit. **Simulation** can sometimes be effective; for example, for prediction problems when a high-quality dynamic theory and precise problem specifications are known (Davis & Marcus, 2014) (Battaglia, Hamrick, & Tenenbaum, 2013). An agent can use **highly trained, specialized manipulations and control regimes,** such as an outfielder chasing a fly ball. **Analogy** is used to relate a new physical situation that has some structural similarities to a known situation, such as comparing an electric circuit to a hydraulic system. **Abstraction** reduces a physical situation to a small number of key relations, for instance reducing a physical electric device to a circuit diagram. **Approximation** permits the simplification of numerical or geometric specification; for instance, approximating an oblong object as a rectangular box. Moreover, all of these modes are to some degree **integrated**; if an outfielder chasing a fly ball and a fan throws a bottle onto the field, the outfielder may alter his path to avoid tripping on it.

Where knowledge of the dynamics of a domain or of the specifications of a situation are extremely weak, the most appropriate reasoning mode would seem to be **knowledge-based reasoning**; that is, a reasoning method in which problem specifications and some part of world knowledge are represented declaratively, and where reasoning consists largely in drawing making inferences, also represented declaratively, from this knowledge. Such forms of representation and reasoning are particularly flexible in their ability to express partial information and to use it in many directions.[3] Our objective in this paper is to present a part of a knowledge-based theory of containers and manipulation.

[3] How knowledge-based reasoning can be implemented in the neural hardware is a difficult problem which we do not attempt to address here. However, we subscribe to the theory (Newell, 1981) that the cognitive processes can be usefully described at the knowledge level at least partly in terms of symbolic representations and symbolic reasoning (Marcus, 2001).

The knowledge-based theory itself has many components at different levels of specificity and abstraction. For example:

- We use a theory of *time* that only involves order relations between instants: time TA occurs before time TB. A richer theory might involve also order relations between durations (duration DA is shorter than DB); or order-of-magnitude relations between durations (DA is much shorter than DB); or a full metric theory of times and durations (DA is twice as long as DB). However, the examples we consider in this paper do not require those.
- Our theory of *spatial and geometrical relations* has a number of different components. For the most part, we use topological and parthood relations between regions, such as "Region RA is part of region RB," "RA is in contact with RB", or "RA is an interior cavity of RB." However we also incorporate a theory of order-of-magnitude relations between the size of regions ("RA is much smaller than RB").
- Our theory of the *spatio-temporal characteristics of objects* includes the relations "Object O occupies region R at time T'', "Region R is a feasible shape for object O" (that is, O can be manipulated so as to occupy R), and "The trajectory of object O between times TA and TB is history H."

In many cases, a concept that is important at an abstract level can only be defined exactly or fully characterized at a more concrete level. For example, the full definition of a "continuously changing region" requires a metric over regions which we do not develop here (see (Davis, 2001).) However, one can assert some of the properties of continuous change; for instance, an object with a continuously changing shape cannot go from inside to outside a container without overlapping the container. Therefore we include the concept of a "continuous history" in the qualitative level even though we do not fully define it.

Another, more complex, example: A key concept in the theory of manipulation is the feasibility of moving an object O from place A to place B. It is sometimes possible to show that this action is infeasible using purely topological information; for example, if place A is inside a closed container and B is outside it, then the action is not feasible. Giving necessary and sufficient conditions, however, is much more difficult. In delicate cases, where one has to rely on bending the object O through a tight passage, reasoning whether it is feasible to move O from A to B or not may require a very detailed theory of the physical and geometric properties both of O and of the manipulator.[4] Moreover, because of the frequency and importance of manipulation in everyday life, non-expert people are implicitly aware of many of the issues and complexities involved, though, of course, they cannot always carry out the physical and geometric reasoning involved with perfect precision and accuracy,

However, at this stage of our theory development, we are not attempting to characterize a complete theory of moving an object, or even of the commonsense understanding of moving an object. Rather, we are just trying to characterize some of the knowledge used in cases where the information is radically incomplete and the reasoning is easy. Therefore, rather than presenting general conditions that are necessary and sufficient, our knowledge base incorporates a number of specialized rules, some stating necessary conditions, and some stating sufficient conditions.

[4] Fully detailed physical and spatial *theories*, such as a formal axiomatization of Newtonian mechanics combined with an axiomatization of Euclidean space and real-valued time, can in principle support inference from radically incomplete *problem specifications* using theorem proving techniques. However, in practice, formulating inferences such as those discussed in this paper from a first-principles axiomatization of Newtonian physics is difficult. In particular, standard axiomatizations of Newtonian mechanics do not include agents; and giving a detailed scientific axiomatic characterization of either a biological or a robotic agent would be a very large undertaking.

The theory that we envision, and the fragment of it that we have worked out, is frankly neither an elegant system of equations nor a system of necessary and sufficient conditions expressed at a uniform level of description. It is much more piecemeal: there are constraints, there are necessary conditions, there are sufficient conditions; but these are not "tight". Some of these are very general (e.g. two objects do not overlap), others quite specialized. Some require only topological information, some require qualitative metric information, some require quite precise geometric information. Nonetheless, we believe that this is on the right track because it seems to address the problem and reflect the characteristics of radically incomplete reasoning much more closely than any alternative.

The theory that we have developed does not conform to any well-defined metalogical framework, along the lines of (Sandewall, 1995) or (Reiter, 2001). Such frameworks, when available, have many advantages: they guide theory construction, guide efficient implementation, and allow the possibility of proving metalogical properties such as consistency or computational complexity. Moreover, in both of these theories, correct frame axioms can be derived nonmonotonically; this was, indeed, one of the major motivations of the design of these theories. However, both Sandewall's and Reiter's framework are largely designed for cases where complete dynamic theories are available. In particular, they each assume that the theory describes all changes to fluents; frame axioms can then be derived by positing that any fluent that is not obliged to change remains the same. Therefore these do not work well with reasoning from radically incomplete information, in which it may be indeterminate whether and how a fluent changes. By contrast, in this paper we state our frame axioms explicitly; but this is appropriate, since the theories are too weak to justify the standard default assumption that a fluent does not change unless there is some action that posits that it changes. Quite the contrary, in one case (axiom M.S.A.1, section 7.5.3) we introduce an axiom positing that an unstable state must always be followed, eventually, by a stable state, with no explanation of the mechanism that brings that about. If we could work out a framework for formulating physical theories that is both systematic and supports flexible inference from radically incomplete information, that might well be advantageous, and we hope to revisit the question in future work; but it is beyond the scope of this paper.

Frameworks such as Sandewall's and Reiter's also have the advantage that one can prove meta-level theorems guaranteeing that the theory is consistent with problem specifications of a particular form. As we will discuss in section 9, consistency theorems can be proved for our theory, but they are almost unavoidably narrower in scope.

The design of this knowledge base must also face the issues of the redundancy of rules and of the level of generality at which rules should be stated. Contrary to common practice in axiomatizing mathematical theories, we have invested little effort in finding a minimal collection of axioms, since for our purposes there is little advantage to that. There remains the question, however, of choosing the level of abstraction at which to state the rules, and our choice may strike some readers as leaning implausibly to the abstract side. The motivation for this is to bring out the commonality in different situations.

Consider, for example, the following three facts:

Fact 1: An object inside a solid closed container cannot come out of the container, even if the container is moved around.

Fact 2: In the situation shown in figure 3, the ball must go through the red region before it can reach the green region.

Fact 3: The water in a tea kettle with the lid on can only come out the spout.

It is certainly possible that a human reasoner is applying three entirely separate rules specific to these particular situations. (Undoubtedly, the way in which reasoning is done varies from one person to another, and also changes developmentally.) However, it certainly seem plausible that often people will use the same knowledge in solving all three problems, that they will think of the three problems in the same way, and, if they are presented with all these problems, they will realize that they are similar. An automated reasoner should do likewise. Note, though, that the specific physics of the three situations are quite different: in fact 1, there is a single moving object that is a closed container; in fact 2, there is a closed container formed by the union of the solid walls with the purely spatial region marked in red; in fact 3, there is a closed container formed by the kettle plus lid plus an imaginary cork in the spout. To formulate a principle that subsumes a;; three cases, therefore, requires the fairly abstract concept of a *history*, a function from time to regions of space, that can move around (needed for facts 1 and 3) but is not tied to a physical object (needed for facts 2 and 3).

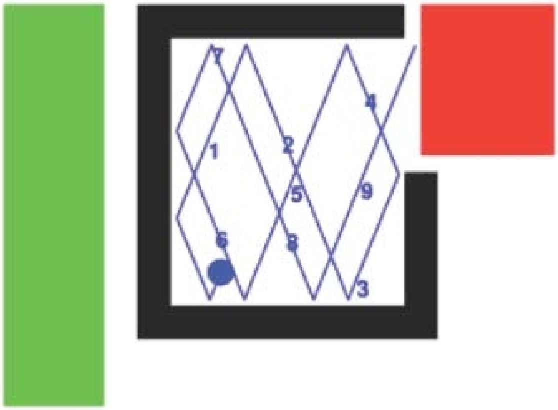


Figure 3: Reasoning about a bouncing ball (from (Smith, Dechter, Tenenbaum, & Vul, 2013))

We use first-order logic with equality as a convenient notation, without at all claiming, either that this is an ideal formalism for an automated system or that it is especially close to cognitive realities. First-order logic has the advantage that it is a standard *lingua franca* (Hayes, 1977), and that there exist standard software inference engines.

3.1 Methodology

Our approach is that of knowledge-based analysis of commonsense reasoning (Hayes, 1979) (Davis, 1998). The results of the analysis, at the knowledge level (Newell, 1981), consists of five parts (figure 4):

1. A *collection of example problems* whose solutions seem commonsensically obvious.
2. A *microworld*. The microworld is a well-defined idealization of the domain, with some limited collection of relations and sorts of entities. The microworld is rich enough to capture the important aspects of the problems in the collection.
3. A *representation language*. We use a first-order language. The meanings of the symbols in the representation language are grounded in the microworld. The representation language is rich enough to express the facts in the knowledge base and to express specifications of the problems in the collection.
4. A *knowledge base,* a formal theory whose meaning is grounded in the microworld and is true in the microworld and that is sufficient to support the inferences needed to solve the problems.

5. *Problem specifications*, expressed in the representation language. The answer to each problem can be justified as an inference, given the problem specification and the knowledge base.

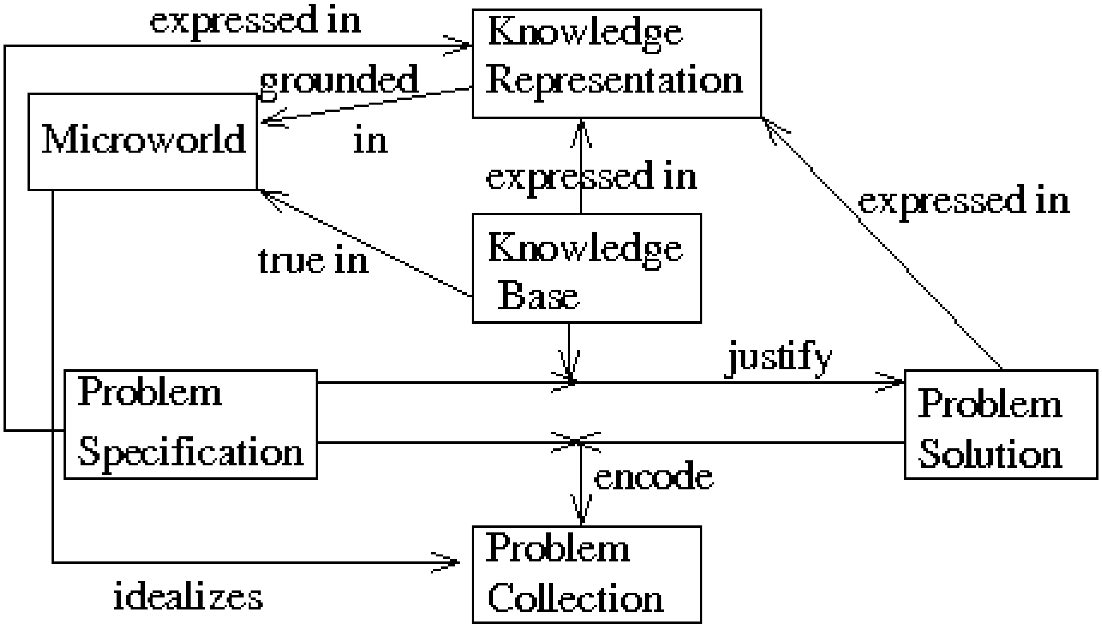


Figure 4: Knowledge-based analysis

In this paper, unlike (Davis, 2011), we require that the axioms be reasonably easily to state in first-order logic. In particular, in the knowledge-based system described here, we avoid the use of *axiom schemas*, infinite collections of axioms, such as the principle of induction or the comprehension axiom from set theory. Axiom schemas are certainly problematic in terms of computational efficiency of inference, and perhaps also in terms of cognitive plausibility.

There are also two further desiderata that we try to achieve for the axioms (these two often conflict, so there is a trade-off to be managed). First, symbols should correspond to concepts that seem reasonably natural in a cognitive model. For instance, ClosedContainer seems plausible; HausdorffDistance, used in (Davis, 2011), seems less so. Second, axioms should be stated at a fairly high-level of generality and abstraction, so that each axiom can be used for many different problems.

For simplicity, we have above portrayed our methodology as sequential: first problems, then microworld, then knowledge representation, then encoding. In practice it is cyclical and iterative. In particular the process of formulating the axioms suggests new problems, improved formulations for old problems, and improvements to the scope and characteristics of the microworld.

Our aim here is not to be comprehensive, but rather to explore basic issues. A complete theory would have to include many additional forms of spatial, physical, and planning knowledge, and would have to integrate other forms of reasoning including simulation, reasoning by analogy, and induction. Nonetheless we believe that our analysis provides insight both into commonsense physical reasoning specifically and into coping with incomplete information generally.

3.2 Evaluation

The difficulties of systematically evaluating such a theory are formidable. As we have argued elsewhere (Davis, 1998), it is in general difficult to evaluate theories of commonsense reasoning in a limited domain, because there is rarely any natural source of commonsense problems limited to a given domain. In the AI literature, the class of commonsense physical reasoning problems that has been studied often reflects what can be easily implemented or what is of immediate practical value; in the cognitive psychology literature (e.g. (Hegarty, 2004) (Battaglia, Hamrick, & Tenenbaum, 2013)) it often reflects the problems that can easily be the subject of controlled psychological experiments. Thus, both directions of research can miss the kinds of problems that people face in ecologically natural settings. The criteria mentioned above in our methodology do not lend themselves to numerical measures of success, and the iterative nature of theory development means that the goal itself is a moving target.

What we have done is to demonstrate that the symbols and rules in the knowledge base are adequate to express and justify simple commonsensical qualitative inferences, discussed below in section 8.

4. From theory to working knowledge base

As we will see in section 7, the theory that we have developed is quite complex, with 6 sorts, 107 other non-logical symbols, 78 definitions and 72 axioms. Moreover the proofs of the sample inferences, in the paper supplement, are long; the proof of inference 4 involves 300 steps. Considering how narrow the scope of the theory is, and how simple the inferences seem, this is rather complex; the reader is certainly justified in wondering how this will scale to richer theories and less obvious inferences. In particular, three questions might leap to mind: How can an automated reasoner be expected to find such long proofs in such a rich theory? How will this handcrafting of knowledge-based theories scale? How can we seriously propose this as a cognitive model? A fourth question, whether it is possible to be sure that the theory is consistent, will be addressed in section 9.

The answer to the first question, regarding the length of the inference chains, is largely that the formulation here is not optimized for automated inference. Rather, the formulation given here is geared toward making comparatively easy for the human reader to read the paper, for the authors to write it, and for both readers and authors to be confident that the symbols are being used consistently and that the axioms are mutually consistent. The axioms have thus the whole been kept minimal and primitive, Also, we have often used many symbols of closely related meaning; this helps readability, but forces the reasoning process to repeatedly go through long chains of definition hunting. In any actual system, many of our lemmas (including, quite possibly, our sample inference 1) would be built in, rather than re-derived each time. Likewise of the defined symbols would probably be replaced by their definitions, to save the labor involved in definition hunting. In short, we would expect that in an implemented knowledge base the chains of reasoning would be shorter than they are here.

Moreover, it should be possible to develop heuristics to focus the reasoning process on key elements. For example, the lengthy proof of inference 4 consists largely of validating frame inferences; proving that, after the robot has carried out a specific action, the objects not involved remain as they were. It may well be possible to systematize the process of inferring these, and thereby reduce the size of the search space.

The second challenge – scaling this up from a toy theory of a hundred axioms to the perhaps hundreds of thousands or millions one would need to cover a large fraction of commonsense knowledge – is of course very real. There are essentially only four solutions on the table: either you use experts to handcraft knowledge bases, or you crowd-source to non-experts, or you use machine learning techniques to derive the knowledge from texts (Davis & Marcus, 2015), or you use machine learning to derive the knowledge from videos or from direct interaction with the world. Hand-crafting by experts is slow and expensive. Crowd-sourcing yields results of very uneven quality, particularly in foundational domains such as spatial, temporal, and elementary physical reasoning. Text-mining is currently highly limited. Learning from seeing the world and interacting with it must ultimately be possible, since it is how children learn about the world, but so far only very preliminary and limited results have been obtained this way (Lerer, Gross, & Fergus, 2016). Our feeling is that, to achieve the desired flexibility in this kind of reasoning, it will be important to analyze what needs to be learned before deploying general purpose learning techniques.

We note a couple of specific points. The work in this paper represents about three person-months of solid work, building on a large body of previous work, and has constructed a theory of about 100 symbols and 150 axioms and definitions, which, we would claim, addresses fundamental issues in physical reasoning and is of quite high quality. If the production scales linearly with the effort, which of course is not at all a safe prediction, then generating a theory of 200,000 axioms would require 250 person years — a large effort but certainly an imaginable one.[5] What fraction of commonsense physical reasoning or of commonsense reasoning generally can be covered in 200,000 axioms is anybody's guess. As a point of comparison, about 1.5 million species of animals have been identified. Each of these identifications was done by hand by a taxonomic biologist (professional or serious amateur) and we presume required not less than a week's work, and often considerably more; and taxonomic biologists are not a dime a dozen. With patience, large projects can be accomplished. While calling for this sort of in-depth of knowledge engineering is outside of today's mainstream, we think it is feasible, and we think it is indispensable.

Finally, with respect to cognitive modeling, our claims are modest.We are only putting this forward as a model at the knowledge level (Newell, 1981) or the computational level (Marr, 1982), not as a process model. All that we would propose is that human reasoners can carry out and do carry out the kinds of reasoning that we are describing; that they would (generally) assent to the correctness of the axioms here, and that doing these kind of reasoning almost certainly requires knowledge and a conceptual apparatus in some ways similar to the theory that we have described, whatever "knowledge" and "concepts" ultimately turn out to be.

At the same time, we do not by any means claim that the set of concepts or the set of axioms presented in this paper is the only correct way to construct a knowledge base or a cognitive theory for this domain. Many of the choices we have made in developing the theory in section 7 are somewhat or entirely arbitrary. We do not suppose that there is a unique right way to construct the knowledge base, or a unique way that different minds think about these issues; rather, there are probably quite a number of ways of constructing a knowledge base that will suffice for these kinds of problems. Rather, the point of this paper is that inference like those discussed in section 8 are important; that previous theories of physical reasoning in the AI and cognitive psychology

[5] It is almost certainly smaller than the amount of effort that has gone into the CYC project (Lenat, Prakash, & Shepherd, 1985).

literature do not address them adequately; but that they can be addressed in a suitably-designed knowledge-based system. The theory presented here is a proof of concept of this last point: With a suitably-designed system, a wide-range of otherwise difficult inferences can be readily captured.

5. "Why Don't You Just Use Simulation?"

The knowledge-based analysis we will propose below is complex, highly incomplete, unimplemented, and untested; completing the theory and producing a reliable implementation are major projects of uncertain success. By contrast, the technology of physics simulators ("physics engines") such as PHYSX is very well established, powerful, and quite general. The reader might reasonably suggest that simulators would be a more promising basis for commonsense physical reasoning than knowledge-based systems.

As we have argued elsewhere, at much greater length (Davis & Marcus, 2016), physics engines, though powerful, are in many ways poorly suited to the needs of commonsense reasoning. In that paper, we analyze a number of features of physical reasoning problems that are inherently difficult for simulation, including incomplete information, unknown physics, irrelevant complexity. Two examples:

1. (Incomplete information and irrelevant complexity). Suppose that you have a closed can, half-way full of sand, and you shake it up and down a few times. You wish to infer that the sand stays in the can. In our knowledge-based approach, that inference is very simple; in fact, it is just an instance of our first sample inference (section 8,1). In a pure simulation approach, it would be necessary to specify, as boundary conditions, the exact shape and initial position of each grain of sand and the exact trajectory of the shaking, and then it would be necessary to trace every collision of two grains of sand together.
2. (Unknown physics) Suppose that you are walking along the beach and you see an oddly shaped mound of green glop. You are wondering what will happen if you kick it. Not knowing what kind of thing it is, you cannot predict that with any precision. Still, there are many scenarios you can rule out; it will not turn into a hummingbird, for example.

More broadly, the seeming greater simplicity of simulation-based theories is partly an illusion due to the familiarity of physics engines and their technology. A state-of-the-art physics engine incorporates an enormous number of sophisticated techniques: for geometric modeling, for motion modeling, for collision detection, and for numerically solving the complex dynamics, which mix differential behavior with discontinuous change. A complete description of such a program would probably be a monograph many times longer than this paper.

There is also an apparent advantage to physics engines in terms of parsimony; against the large number of rules we propose, feasibility can in many instances simply be computed seemingly (to the end user) using computational techniques that apply very generally. However actual physics engine have built in all kinds of assumptions of how things can be described, with all kinds of special cases for how they interact. Even ensuring that the shape description for an object remains topologically coherent as the object moves around (i.e. that the boundary neither develops gaps nor intersects itself) is a challenging problem in many standard shape representations. We would argue that the advantage in parsimony is more apparent than real.

In parallel to this, physics engines might superficially seem more psychologically plausible; in inference 4 below, 300 steps are required to infer that, under suitable circumstances, an agent can drop a small object into an open container and then pull his hand out, leaving the object in the container. But if one were to look at a full trace of what is happening in a physics engine simulating an instance of this process, that would also look implausible as a cognitive theory. The explanation, in both cases, is partly that what is happening in physical and spatial reasoning below the level that is accessible to conscious introspection must be more complicated than one might suppose; and partly that both of these theories are accounts at the computational level, not the algorithmic level. At present, *both* theories lack sufficient psychological grounding, but then again neither can yet be ruled out on psychological grounds, either, since our knowledge about how computational level theories are algorithmically realized (and realizable) remains primitive.

Of course, it is a fact that a powerful theory of simulation now exists[6] and the technology is implemented. That fact is of very great practical importance if one wishes to build an AI physical reasoning engine over the short term. However from the point of view of building, over the long term, an AI system capable of general physical reasoning, and still more from the point of view of developing a cognitive model of physical reasoning, that fact that, in 2016, existing physics engines are powerful and sophisticated and logic-based qualitative physical reasoners are not, may be largely a historical happenstance.

The more important point is this. It is easy to look at the collection of particular cases that are individually described in our theory, to contrast this with the broad scope of the physical theories that underlie a physics engine and to conclude that the scope of a physics engine is enormously greater than the scope of our theory. After all, a physics engine for solid rigid objects can handle all kinds of physical phenomena that we have not begun to characterize: projectiles, gyroscopes, collisions, sliding, rolling, and so on. What is easily missed, though, is that our theory can deal with all kinds of inferences that a simulation-based physics engine cannot. First, as discussed in section 4.4, since our inferences are monotonic, they are valid whatever additional facts are true about the situation and whatever else is happening.

Second, our inferences apply generally, across broad classes of objects. For instance, in a physics engine, if you want to reason about manipulation by a robot, human, or animal, you need to create a physical model of the interactions of the agent with the outside world; each new type of agent requires a new model. There is, in fact, a small cottage industry in building such models and building infrastructure for them. Our model of manipulation is much less precise and more limited in terms of the kinds of manipulations it describes, but it applies without change across a broad range of agents.

Third, in a logic-based system, any two logically equivalent inferences have essentially the same proof; and therefore the same reasoning system can be used for inferences in very different directions. For example, the inference in Scenario 6.1 states that if ob is a rigid object and a closed container and contains object os at time ta then ob contains object os at any later time; this is a prediction problem. But the same proof will show that if ob is rigid and does not contain object os at time tb, then it does not contain object os at any earlier time, which is a postdiction

[6] Though not all the issues in the physics or in the mathematics have been resolved, even for the case of rigid solid objects (Stewart, 2011).

problem. It also shows that if object ob contains os at time ta, and does not contain os at a later time tb, then ob is not rigid; this is a problem of inferring object characteristics from observations over time. In a simulation-based reasoner, inferences other than prediction are problematic, and certainly these equivalences do not hold. One approach that would sometimes work would be to first run a logic-based front end that translates a non-predictive problem into a prediction problem; then run a simulator to do the prediction; then run a logic-based back-end to translate the answer to the prediction problem into a solution to the original. However, this is not a general solution.

Finally, if one considers the problem of commonsense physical reasoning in the larger context of implementing commonsense knowledge generally, rather than as in isolation, the knowledge-based approach seems much less anomalous. Among various forms of reasoning, physical and mathematical reasoning are almost alone in having elegant, comprehensive theories that often lend themselves to highly efficient specialized algorithms. In most areas of commonsense reasoning, as far as anyone knows, one is necessarily faced with the task of organizing a large, amorphous body of knowledge, with no overarching elegant theory. From that point of view, the kind of theory described here seems very much what one would expect; unusual only in that many of the axioms actually can be justified in terms of standard theories of geometry and physics.

6. Preformal sketch of the microworld and the inferences

In this section, we will present a preformal description of the microworld that we have in mind, and sketch some of the characteristics of the theory. In section 7, we will present a full formal account of the microworld and the knowledge base.

The physical world consists of a collection of objects, which move around in time over space. Objects are distinct; that is, one object cannot be part of another or overlap spatially with another. They are eternal, neither created nor destroyed. They move continuously. An object occupies a region of some three-dimensional extent (technically, a topologically regular region); it cannot be a one-dimensional curve or two-dimensional surface. Objects can be flexible and can change shape, but we do not consider cutting an object into pieces to make several objects or gluing multiple objects together to make a single object. We assume that an object occupies an interior connected region; that is, it does not consist of two parts only connected at a point or along a one-dimensional curve.

This object ontology works with solid, indestructible objects. It does not work well for liquids, thought it does not entirely exclude them;[7] ontologies for liquids are developed in (Hayes, 1985) and (Davis, 2008).

For any object O, there is some range of regions that O can in principle occupy, consistent with its own internal structure; these are called the *feasible* regions for O. For instance, a rigid object can in principle occupy any region that is congruent (without reflection) to its standard shape. A string can occupy any tube-shaped region of a specific length and diameter. A particular quantity of liquid can occupy any region of a specific volume.

6.1 Containers

[7] A liquid can be modeled in this ontology as a collection of drops, each of which is a separate object.

We are primarily concerned with containers and their contents. We distinguish four particular types of containers (figure 5).

- A *closed container* is an object or set of objects that completely envelopes an internal cavity.
- An *open container* surrounds a cavity on all sides but one, where it has a single opening.
- An *upright open container* is an open container with the opening on top.
- A *box with lid* is a pair of objects that together form a closed container for a cavity, and that have the property that, if the box is moved, the lid will remain in place.

The formal theory of these relations is given in sections 7.3.2, 7.4.3, and 7.4.4. "Closed container", "open container" and "open upright container" are defined purely in terms of the geometry of the objects involved. "Box with lid" involves both geometrical and physical characteristics, since the constraint that the lid will remain on the box depends on the physical characteristics of the two objects.

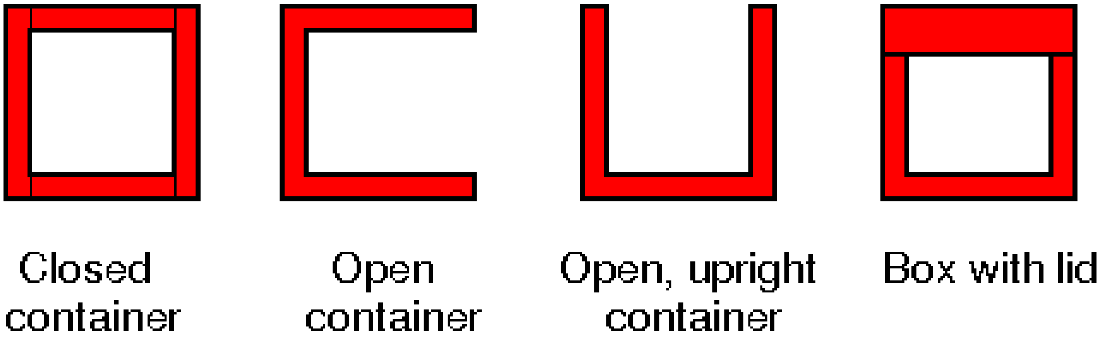


Figure 5: Types of containers

In a container made of flexible material, cavities can split and merge; they can open up to the outside world or close themselves off from the outside world.[8]

To characterize cavities dynamically, we use *histories*; that is, functions from time to regions (Hayes, 1979). Thus the value of history H at time T is a region denoted Slice(T,H). The place occupied by an object, or by a set of objects, over time is one kind of history. We say that a history Hc is a *dynamic cavity* of history Hx from time Ta to time Tb if it satisfies these two conditions:

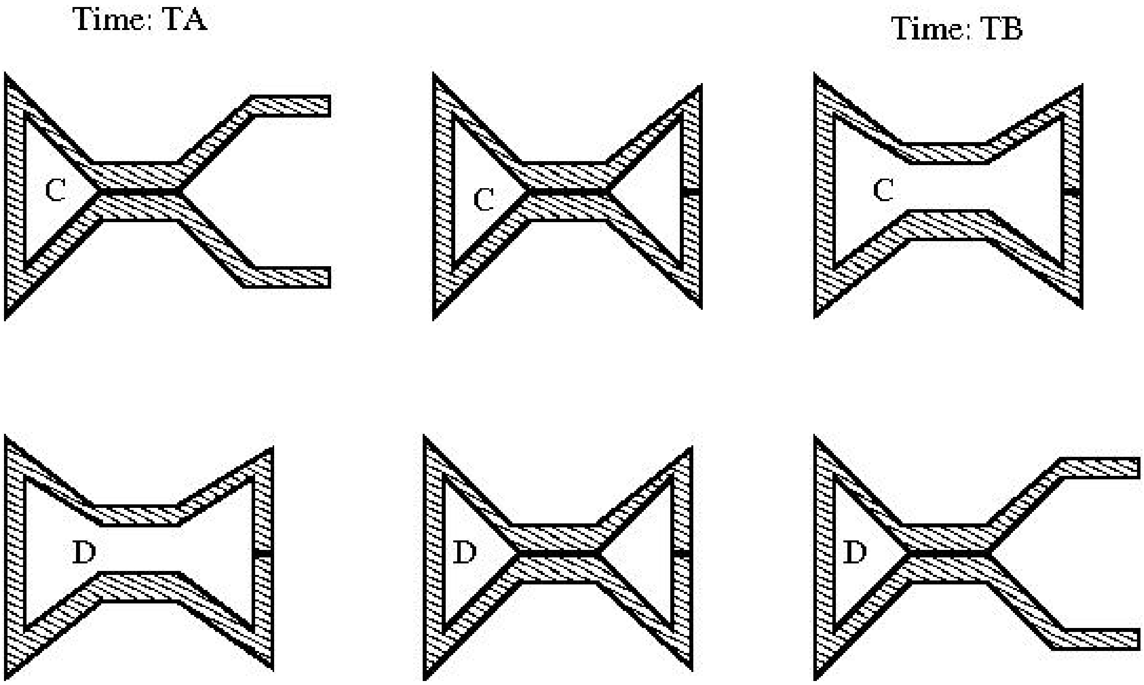
- At all times Tm strictly between Ta and Tb, Slice(Tm,Hc) is a cavity inside the spatial closed container Slice(Tm,Hx).
- Hc is *weakly continuous*. That is, for any time Tm there exists an interval (Tc,Td) and a region R such that throughout (Tc,Td) R is part of Hc. Intuitively, a cavity is weakly continuous if a small marble that can foresee how Hc will evolve and can move arbitrarily quickly can succeed in staying inside Hc.

We distinguish three categories of dynamic cavities (figure 6):

- Hc is a *no-exit cavity* of Hx if there is no way to escape from Hc except through Hx.
- Hc is a *no-entrance cavity* of H if there is no way to get into Hc except through Hx.

[8] The classic discussion of cavities and in particular the individuation of cavities is (Casati & Varzi, 1994).

- Hc is a *persistent cavity* of H if it is both a no-exit and a no-entrance cavity.



C is a no-exit cavity from TA to TB.
D is a no-entrance cavity from TA to TB.

Figure 6: Dynamic cavities

6.2 The agent and his actions

There is a single agent, who himself is an object. The agent is capable of moving by himself, grasping other objects, manipulating other objects while grasping them, and releasing them. The theory developed here of these actions, particularly grasping and manipulating, is very weak. We specify some general necessary geometric conditions for being able to grasp an object (e.g. the agent must be geometrically touching the object) but no sufficient conditions. The feasibility of grasping an object therefore has to be stated as part of the problem specifications. We assume that the agent can release an object he is grasping at any time. If the object is in a stable position when he releases it, it will stay where placed; if not, it will fall.

Manipulating one object may cause other objects to move as well. In some cases this effect can be predicted; e.g. if the agent is manipulating a closed container, then the objects inside the container move along. In some cases one can predict that moving one object will not cause another to move, if they are parts of two sets that are causally isolated from one another (this is discussed further in section 6.3). Otherwise, our theory leaves the effect of moving one object on another indeterminate. We do posit that that the agent can only *directly* cause the motion of an object by manipulating it, rather than by pushing it or hitting it; that is, we assume that the agent is careful not to push or hit a movable object, if he is not deliberately manipulating it. Thus, in our theory, the agent can manipulate object A, which pushes on object B, with largely indeterminate effect but it cannot directly push on object B with its own body.

Our causal theory thus includes two kinds of actions: the agent can travel empty-handed to region r, represented by the *Event* Travel(r), or the agent can manipulate an object by grasping it, moving it, and releasing it. Section 7.5 develops weak causal theories of travelling and ungrasping. Richer theories of causation and of physical preconditions are associated with more specialized actions. In section 7.8, we develop a theory of one such specialized action: loading an object into an open upright container.

6.3 Physical laws

The qualitative theory of physics developed here is divided into six parts: general physical laws; basic laws of the agent's motion and manipulation; a theory of open containers; a theory of stability and falling; frame axioms, which limit the kinds of changes that occur; and specialized axioms for specific actions. We sketch each of these here; full details are given in section 7.

General physical laws.

- Two distinct objects do not overlap spatially.
- The trajectory of an object is a continuous function of time.
- An object o occupies a region feasible for o.

Motion and manipulation

- The agent can grasp an object o only if he is spatially in contact with o. The agent can manipulate o only if he is grasping it.
- If the agent is holding an object, he can release it at any time.

Open containers

- If an object is in an upright open container, and the agent moves the container and keeps it upright, then the object will remain in the container.
- If a lid is properly placed on a box, and the agent moves the box and keeps it upright, then the lid will stay on the box.

We do not need comparable physical axioms for closed containers; the fact that an object necessarily remains in a closed container is a consequence of the general physical laws, that objects move continuously and do not overlap, together with spatio-temporal theorems.

Stability and falling

- If an object is not being grasped and is in an unstable position, it will fall for a time, then reach a stable position.
- If an object is inside an upright open container and falls, it will remain inside the container.

We do not give any geometric rules for evaluating stability. The theory in this paper is consistent with the possibility that an object is stable while floating in mid-air. A stronger theory of stability is given in (Davis, 2011).

Frame axioms: The most important of the frame axioms governs change in position, and is formulated in terms of "causally isolated'' sets of objects. A set of object ox is isolated by a set of objects os if the agent cannot cause any of the objects in ox to move without moving some of those in os.

- At any time, if the agent is not manipulating any object and all objects are stable, then no object except the agent moves. (This corresponds to a quasi-static physics, in which dissipative forces like friction are large enough to rapidly stop any inertial motion.)
- If a set of objects s is causally isolated during the interval [ta,tb] and all the objects in s are in a stable position at time ta, then all the objects in s remain motionless and stable throughout the interval.

In the real world, there are, of course, exceptions to many of these rules; but in everyday settings they are generally true or approximately true.

Specialized action:

- If the agent can both grasp object o and reach the inside of open container oc and if the current contents of oc together with o are small as compared to the inside of oc, then the agent can load o into oc.

6.4 Sample inferences

In section 8 we establish the power of the theory constructed in section 7 by showing that it suffices to justify a collection of sample inferences.

1. An object inside a rigid closed container remains inside.
2. If object oa is inside a closed container ob, which is inside a closed container oc, and oc is a rigid object, then oa remains inside oc.
3. In the problem shown in figure 3, the ball must reach the red region before it can reach the green region.
4. If object ox4 is outside upright container ob4, and the current contents of ob4 together with ox4 are much smaller than the interior of ob4, and the agent can reach and move ox4 and can reach into ob4, then the agent can load ox4 into ob4.
5. Let ob5 and ol5 be a box with lid at time ta5, and let os5 be an object inside the box. Assume that the agent is outside the box at time ta5. If os5 is somewhere else at time tb5, and the box is fixed throughout [ta5,tb5] then the lid must have moved at some time in between ta5 and tb5.

Two key features of these inferences should be noted. The first is the weakness of the information provided. In none of these do we require any geometric specifications of the objects involved, except the containment relation. The inferences do require that some of the objects are rigid; but the objects that are not constrained to be rigid can be anything at all.

The second feature is that the inferences remain valid *whatever else is true and whatever else is going on.* In inference (1) above, for example, it may be that the container is a box containing fifty randomly shaped objects of unknown physical characteristics, and that the box is being tossed up and down. Nothing in the problem specification rules this out. The conclusion is still valid; in fact, the proof is unchanged. (It may, of course, be more difficult for an automated reasoner to find the proof, amid this distracting additional information.) In inference 4, ox4 may itself be a closed container with objects inside, or an upright open container, or a box with a lid. This is one of the major advantages, often overlooked in the AI literature, of deductive inference as opposed to plausible inference. In any system of plausible inference, adding any additional fact can potentially upset the entire applecart; you always have to check that the new fact does not disrupt assumptions you have made. In non-monotonic logic, you have to check that the new fact does not contradict default assumptions (weaken circumscriptive axioms etc.). In probabilistic reasoning, you have to check that the old conclusion is independent of the new information. By contrast, in monotonic logic, a valid proof from premises remains valid, whatever else is learned.

7. Formal theory of the microworld and the knowledge base

We now proceed to the formal account of the microworld, the representation language, and the knowledge base. In section 8, we will present formal specifications for some problems.

7.1 Logic, Sorts, Notation

The representation language is a sorted (typed) first-order logic with equality. We use symbols in lower case Arial font for variables, such as u,v; symbols in Arial font, starting with an upper case character, for constants, function, and predicates symbols; such as Lt or Union; and symbols in italics for sorts, such as *Time* or *Region*.

The sorting system is simple.

- A sort is equivalent to a monadic predicate.
- The space of entities is partitioned into 6 disjoint sets: *Time, Region, History, Object, ObjectSet* and *Action*.
- Every non-logical symbol (constant, function, predicate) has a unique sortal signature. We do not use overloading or polymorphism.

The precedence of Boolean operators is: ¬, ∧, ∨, ⇒, ⇔. The quantifiers ∀, ∃, and $\exists^1$ (unique existence) have scope until the end of the formula or close bracket of larger scope.

The entities in the universe are partitioned into sorts. Each entity is of exactly one sort. There are six sorts: *Time, Region, History, Object, ObjectSet* and *Event*. For each sort there is a corresponding unary predicate, written in typewriter font; for example, the predicate Time(t) corresponds to the sort *Time*.

We use italicized sortal symbols in two contexts. The first use is for restricted quantification.[9] A quantified variable can be restricted to a sort, with the standard meanings: If μ is a variable, α is a sortal symbol and ϕ(μ) is a formula, then

$\forall_{\mu:\alpha}\, \phi(\mu)$ is equivalent to $\forall_{\mu}\, \alpha(\mu) \Rightarrow \phi(\mu)$ and

$\exists_{\mu:\alpha}\, \phi(\mu)$ is equivalent to $\exists_{\mu}\, \alpha(\mu) \wedge \phi(\mu)$.

For example

$\forall_{u,v:\mathit{Time}}\, \mathrm{Leq}(u,v) \Leftrightarrow \mathrm{Lt}(u,v) \vee u=v$ is equivalent to

$\forall_{u,v}\, [\mathrm{Time}(u) \wedge \mathrm{Time}(v)] \Rightarrow [\mathrm{Leq}(u,v) \Leftrightarrow \mathrm{Lt}(u,v) \vee u=v]$

The second use of sortal symbols is in the declaration of non-logical symbols. Every non-logical symbol is introduced with a declaration of the sorts of its arguments and values. In our theory, sorting of non-logical symbols is strict; every symbol except the equality and inequality signs is sorted and there is no overloading or polymorphism. Each such declaration implicitly expresses a sortal axiom governing the symbol. The syntax of declarations is modeled on the syntax of function declarations in programming languages such as Pascal or Ada. These declarations and axioms are of three types:

[9] We include sortal specifications in the quantifier only when it is necessary; i.e. the formula would be false if the sortal specification were omitted.

- **Constant symbols.** A constant symbol has a declaration of the form Symbol → *Sort*. The corresponding axiom states that the symbol is of the sort. For example, the declaration "Ta → *Time*" corresponds to the axiom "Time(Ta)".

- **Predicate symbols.** A predicate symbol declaration declares a sort for each argument. The corresponding axiom asserts that the predicate holds on arguments only if the arguments are of the proper sorts. For instance, the declaration
  "Continuous(ta,tb: *Time*; h:*History*)" corresponds to the axiom
  "$\forall_{ta,tb,h}$ Continuous(ta,tb,h) ⇒Time(ta) ∧ Time(tb) ∧ History(h)."

- **Function symbols**. A function symbol declaration declares the sorts of each argument and the sort of the result. The corresponding axiom asserts that if the arguments have the specified sorts, then the result has the specified sort. For example, the declaration
  "Slice(t:*Time*; h:*History*) → *Region*" corresponds to the axiom
  "$\forall_{t,h}$ Time(t) ∧ History(h) ⇒ Region(Slice(t,h))."
  Functions are all total over the space of arguments of the proper sort. (Presumably, the function is undefined if the sortal conditions on the arguments are not met, but we do not axiomatize that.)

Thus, the entire theory in the sorted logic can be translated into an equivalent theory in an unsorted logic

We have not formalized the distinction between a definition and an axiom, but what we intend is that a definition of symbol s allows s to be replaced by its defining term in all meaningful contexts. How that is achieved depends on what s is. For instance, a predicate is defined by necessary and sufficient conditions; a set is defined by specifying necessary and sufficient conditions on its elements; an event is defined by specifying necessary and sufficient conditions for its occurrence; and so on. A definition of a constant or a function symbol in terms of a property can be a substantive axiom, since it implies that an entity satisfying that property exists. For instance, in section 7.4.1, we define the function Pair(oa,ob) as a function mapping two objects oa and ob to the set {oa, ob}; this definition corresponds to the usual Zermelo-Frankel axiom of pairing, since it entails that such a set exists.

The axioms here are *sufficient* to prove the five sample inferences of section 8, and they are *necessarily* true in our intended model. However neither converse holds: some of the axioms are not used in any of the sample inferences and thus, not necessary for those inferences; and the axioms are not sufficient to enforce all the properties of the model described in the text. For instance, as regards the axioms of time, it turns out, somewhat surprisingly, that the only proper axiom of time that we use in our inferences, is T.I.A.2, transitivity; that is, our inferences will work in any model of time in which "earlier than" is transitive. In the other direction, the axioms certainly do not suffice to enforce the condition that time lines are continuous (isomorphic to the real line.) The choice of which axioms to include here is determined, partly by considerations of which axioms we think might be used in other commonsense reasoning, partly by the aesthetics of axiomatization. It would seem too weird to omit anti-symmetry in an axiomatization of time, even though this property never happens to come up in our sample inferences.

We organize our presentation in this section in terms of subtheories in a dependency order. The subtheories are organized in a two-level hierarchy. The structure of this section of the paper corresponds to this same hierarchy: Top-level subtheories correspond to subsections of the paper (e.g. Space is subsection 7.3) and lower-level subtheories correspond to sub-subsections (e.g. Containment is sub-subsection 7.3.2). In each section or subsection, we first declare the new formal symbols introduced, then enumerate the definitions, then enumerate the axioms. The symbols used in the definitions and axioms for a given subtheory are introduced either in that subtheory or in previously presented subtheories; that is, no subtheory depends in any way on material presented later.

Axioms are numbered using four field designators. The first two indicate the second and subsection; the third is `D' or `A' for definition or axiom; the fourth is just an enumerative number. Occasionally there may be redundant axioms; unlike a mathematical context, in developing a knowledge base, eliminating redundancy is not in general a priority.

Top-level	Lower-level	Section	Scenario
Time (T)		7.2	
	Ordering (T.I)	7.2.1	All
	Actions (T.A)	7.2.2	4
Space (S)		7.3	
	Basic (S.B)	7.3.1	All
	Containment (S.C)	7.3.2	All
	MuchSmaller (S.M)	7.3.3	4
Object (O)		7.4	
	Object Sets (O.S)	7.4.1	4,5
	Spatio-Temporal (O.T)	7.4.2	All
	Objects contain regions (O.R)	7.4.3	All
	Objects contain objects (O.C)	7.4.4	All
	Fits and small set (O.F)	7.4.5	4
	Isolates (O.I)	7.4.6	5
Manipulation (M)		7.5	
	Grasp (M.G)	7.5.1	4,5
	Motion (M.O)	7.5.2	4,5
	Stability and falling (M.S)	7.5.3	4,5
	Frame axioms (M.F)	7.5.4	4,5
	Feasibility of Travelling (M.T)	7.5.5	4
Histories (H)		7.6	
	Basic History (H.I)	7.6.1	All
	Dynamic containers (H.C)	7.6.2	All
	Dynamic upright container (H.U)	7.6.3	4
Rigid Objects (R)		7.7	
	Basic Rigid Objects (R.O)	7.7.1	1,2,5
	Box with Lid (R.B)	7.7.2	4,5
Actions (A)		7.8	
	Safe Manipulation (A.S)	7.8.1	4
	Loading an Upright Container (A.L)	7.8.2	4

Table 1: Subtheories

Table 1 enumerates the subtheories, the character designators, the corresponding paper sections, and the scenarios that illustrate the use of the subtheory. Figure 7 displays the dependency relations between the subtheories. Both the division into the lower-level subtheories and the dependencies between subtheories are substantially arbitrary and should not be taken very seriously.

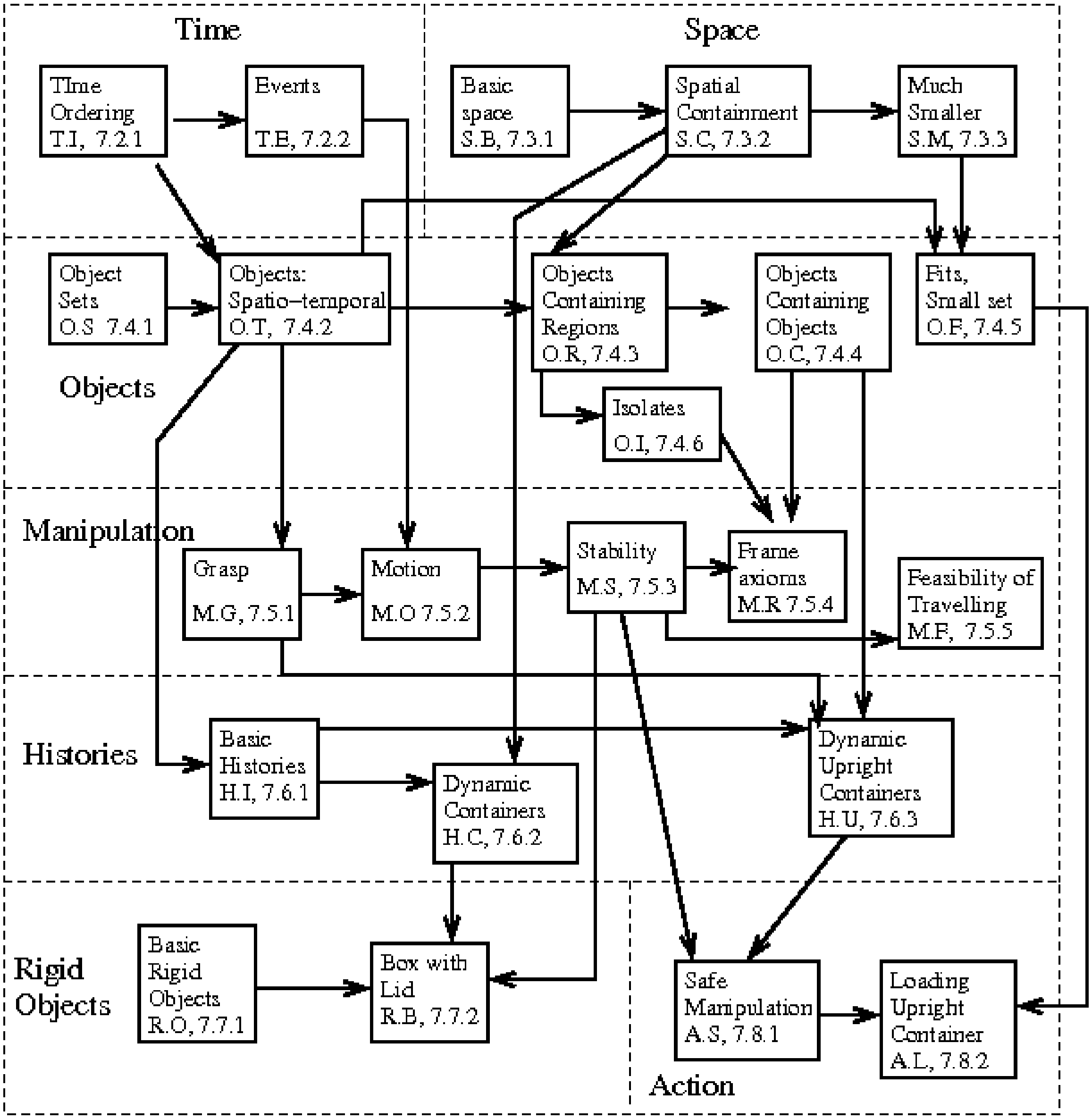

Time
Space
TIme Ordering T.I, 7.2.1
Events T.E, 7.2.2
Basic space S.B, 7.3.1
Spatial Containment S.C, 7.3.2
Much Smaller S.M, 7.3.3
Object Sets O.S 7.4.1
Objects: Spatio–temporal O.T, 7.4.2
Objects Containing Regions O.R, 7.4.3
Objects Containing Objects O.C, 7.4.4
Fits, Small set O.F, 7.4.5
Isolates O.I, 7.4.6
Objects
Manipulation
Grasp M.G, 7.5.1
Motion M.O 7.5.2
Stability M.S, 7.5.3
Frame axioms M.R 7.5.4
Feasibility of Travelling M.F, 7.5.5
Histories
Basic Histories H.I, 7.6.1
Dynamic Containers H.C, 7.6.2
Dynamic Upright Containers H.U, 7.6.3
Rigid Objects
Basic Rigid Objects R.O, 7.7.1
Box with Lid R.B, 7.7.2
Safe Manipulation A.S, 7.8.1
Loading Upright Container A.L, 7.8.2
Action

Figure 7: Dependencies among subtheories

7.2 Time

In our intended model, time is forward-branching and continuous; each maximal fully-ordered path through the time structure is isomorphic to the real line ℝ. Forward branching corresponds to an agent's choice between actions. Branches occur *after* instants; that is, an interval that is bounded and open on the right has a unique least upper bound, but there can be any number of non-overlapping intervals with the same lower bound. For instance, in figure 8 below, the figure on the left shows a permissible branching, in which the open interval U has a single end point B, and the closed intervals [B,C], [B,D], and [B,E] have the common start point B and are otherwise disjoint. The figure on the right shows a non-permissible topology for branching, in which the open interval U is followed by three possible endpoints, B1, B2, and B3; and the closed intervals [B1,C], [B2,D], and [B3,E] each has a different starting point.[10]

The axioms of time given below do not enforce these conditions, but the conditions are assumed in theories developed later in the paper. For instance, as discussed in section 7.5.1, it is assumed that a grasping or non-grasping relation holds over a time interval that is open on the left and closed on the right; that choice is arbitrary in physical terms, but it is made in order to conform to this model.

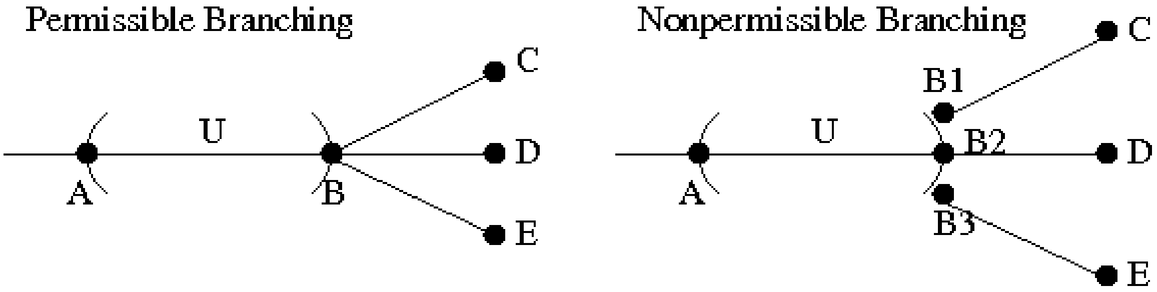


Figure 8: Permissible and nonpermissible topologies of time branching

Since time is forward branching, it is not totally ordered; but the times previous to any given time Z are totally ordered (axiom T.I.A.4 below).

It will be convenient to view the time structure as containing all possible configurations of the objects. This allows us to define the predicate FeasiblePlace(o,r), (region r is a feasible configuration for object o) as true if and only if there is some time when r is the region occupied by o, and to define the predicate Fits(s,r), (object set s fits inside r) as true if and only if there is some time when s is inside r (section 7.4.5).

As a convention, in any predicate or function where there are both temporal arguments and arguments of a different sort, we place the temporal arguments first.

[10] In topological terminology, the right-hand figure is a non-Hausdorff topology; these are generally considered somewhat pathological.

7.2.1 Time Ordering

Symbols:
Lt(x,y: *Time*) —Time x is earlier than time y.
Leq(x,y: *Time*) —Time x is earlier than or equal to time y.
Ordered(x,y: *Time*).
Leq3(x,y,z: *Time*). $x \leq y \leq z$.

Definitions:
T.I.D.1 $\forall_{x,y:Time}$ Leq(x,y) $\Leftrightarrow$ Lt(x,y) $\lor$ x=y.
T.I.D.2 $\forall_{x,y:Time}$ Ordered(x,y) $\Leftrightarrow$ Leq(x,y) $\lor$ Leq(y,x).
T.I.D.3 Leq3(x,y,z) $\Leftrightarrow$ Leq(x,y) $\land$ Leq(y,z).

Axioms:
T.I.A.1 $\neg$[Lt(x,y) $\land$ Lt(y,x)]. Lt is antisymmetric.
T.I.A.2 Lt(x,y) $\land$ Lt(y,z) $\Rightarrow$ Lt(x,z). Lt is transitive.
T.I.A.3 Lt(x,y) $\Rightarrow \exists_z$ Lt(x,z) $\land$ Lt(z,y). The time line is dense.
T.I.A.4. Lt(x,z) $\land$ Lt(y,z) $\Rightarrow$ Ordered(x,y).
Forward branching: The times earlier than z are totally ordered.

7.2.2 Actions

An action a is executed over an extended interval [ta,tb] where ta < tb. An action a is feasible at time t if it is executed on some branch of the time line starting at t. The action Sequence(a1,a2) occurs if a1 and a2 are executed in sequence.

Symbols:
Occur(ta,tb: *Time*; a: *Action*).
Feasible(t: *Time*; a: *Action*).
Sequence(a1,a2: *Action*) → *Action.*

Definition:
T.A.D.1 Feasible(ta,a) $\Leftrightarrow \exists_{tb}$ Occurs(ta,tb,a).

T.A.D.2 $\forall_{ta,tb:\ Time;\ a1,a2:\ Action}$ Occurs(ta,tb,Sequence(a1,a2)) $\Leftrightarrow$
$\exists_{tx}$ Occurs(ta,tx,a1) $\land$ Occurs(tx,tb,a2).

Axiom:
T.A.A.1 Occurs(ta,tb,a) $\Rightarrow$ Lt(ta,tb).

7.3 Spatial Relations

Space is assumed to be three-dimensional Euclidean space $\mathbb{R}^3$. As it happens, none of the axioms in this paper depend very strongly on that assumption; they certainly all hold in $\mathbb{R}^k$ for any $k > 1$, and probably in less realistic spatial models as well (e.g. carefully constructed discrete models of space.) A Region is a bounded, topologically regular, set of points in $\mathbb{R}^3$.

7.3.1 Basic spatial relations

We use the RCC-8 (Randell, Cui, & Cohn, 1992) binary spatial relations P, C, O, DR, EC, DC and OV. The definitions and axioms S.B.A.2 and .3 below are standard in the RCC literature. S.B.A.1 asserts, in effect, that the RCC relation EQ is in fact logical equality. Axiom S.A.B.4 asserts that if region w overlaps the union of u and v then it overlaps either u or v For an extensive discussion of the axiomatization of RCC, see (Pratt & Schoop, 1998) (Pratt-Hartmann, 2007).

Symbols:

P(u,v: *Region*) — Region u is a subset of v.
C(u,v: *Region*) — Regions u and v are in contact.
O(u,v: *Region*) — Regions u and v overlap.
DR(u,v: *Region*) —Regions u and v do not overlap.
EC(u,v: *Region*) — Regions u and v are externally connected.
DC(u,v: *Region*) — Regions u and v are disconnected.
OV (u,v: *Region*) — Regions u and v partially overlap.
RUnion(u,v: *Region*) → *Region* — Union of regions u and v.

Definitions:

S.B.D.1 $\forall_{u,v:\,Region}\ P(u,v) \Leftrightarrow \forall_{w}\ C(w,u) \Rightarrow C(w,v).$

S.B.D.2 $O(u,v) \Leftrightarrow \exists_{z}\ P(z,u) \land P(z,v).$

S.B.D.3 $\forall_{u,v:\,Region}\ DR(u,v) \Leftrightarrow \neg O(u,v).$

S.B.D.4 $EC(u,v) \Leftrightarrow DR(u,v) \land C(u,v).$

S.B.D.5 $\forall_{u,v:\,Region}\ DC(u,v) \Leftrightarrow \neg C(u,v).$

S.B.D.6 $\forall_{u,v:\,Region}\ w{=}RUnion(u,v) \Leftrightarrow$

$$P(u,w) \land P(v,w) \land \forall_{x}\ P(u,x) \land P(v,x) \Rightarrow P(w,x).$$

S.B.D.7 $OV(u,v) \Leftrightarrow O(u,v) \land \neg P(u,v) \land \neg P(v,u)$

Axioms:

S.B.A.1 $P(u,v) \land P(v,u) \Rightarrow u{=}v.$

S.B.A.2 $\forall_{u:\,Region}\ C(u,u).$

S.B.A.3 $C(u,v) \Rightarrow C(v,u).$

S.B.A.4 $O(u,RUnion(v,w)) \Rightarrow O(u,v) \lor O(u,w).$

7.3.2 Spatial Containment

For convenience, we define multiple symbols for what are essentially the same containment relations applying to one region containing another (this section); to an object or set of objects containing a region (section 7.4.3); or to one object or set of objects containing another (section 7.4.4).

Region R is a *closed container* for cavity C (a region) if C is an interior-connected, bounded component of the complement of R (figure 9).

Region R is an *open container* for cavity C if there exists a region A between two parallel planar surfaces S1 and S2 such that:

- A and R do not overlap. The intersection where they meet R ∩ A is (in three dimensions) equal to the ring around A separating S1 and S2: R ∩ A = Bd(A) \ (S1∩S2).

- C is a cavity of the union R ∪ A, but is not a cavity of either R or of A separately.

Region R is an *upright open container* for cavity C if the planar surfaces S1 and S2 associated with A are horizontal and A is above C.

Region R is a *simple upright open container* for cavity C if C is the unique maximal interior with respect to which R is an upright open container (figure 10)

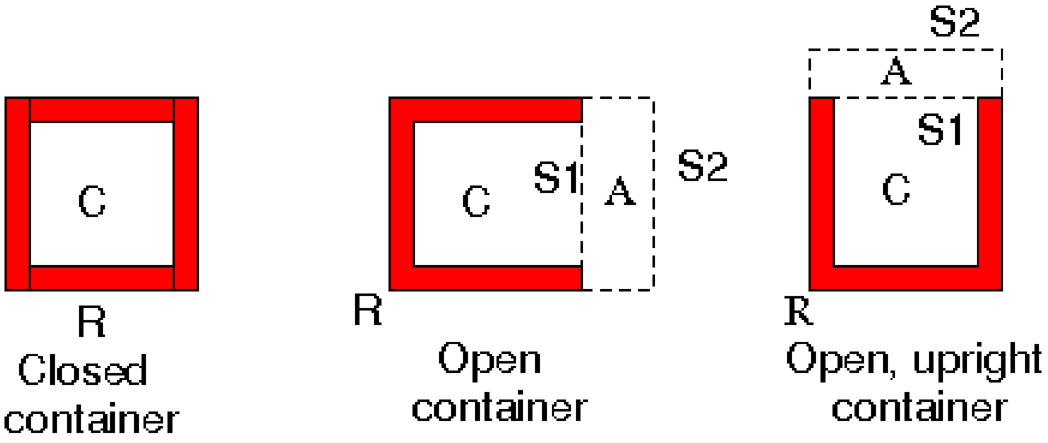
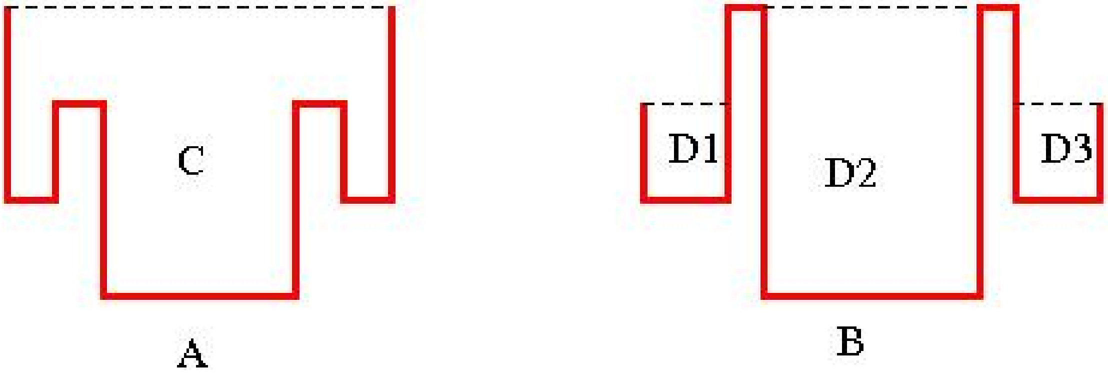


Figure 9: Closed, open, and upright containers



A is a simple upright container: C is the unique maximal region contained.
B is an upright container that is not simple; D1, D2, D3 are each maximal contained regions.

Figure 10: Upright Containers

The definition of closed container is purely topological, and therefore is expressible in our qualitative spatial language. However, expressing the conditions that the surfaces S1 and S2 are planar and parallel would require a more powerful geometric theory than we are undertaking here. We therefore leave OpenContainerShape as a primitive.

We also define the function ConvexHull(r) and the relation FullyOutside(ra,rb), which is defined as holding if the convex hulls of ra and rb are disconnected (definition S.C.D.7). The relation FullyOutside is useful as a sufficient condition to establish that one object does not contain another and that two objects do not interact. We posit a few useful axioms of ConvexHull.

Symbols:
IntConn(r: *Region*). —Region r is interior connected.
Cavity(u,v: *Region*) —Region u is an interior cavity of v.
Outside(u,v: *Region*) —Region u is outside region v.
(u is a subset of the unbounded connected component of the complement of v).
Contained(u,v: *Region*). —Region u is is inside a cavity in v.
CombinedContainer(ra,rb,rc: *Region*). — Region rc is an interior cavity of ra ∪ rb.
OpenContainerShape(rb,rc: *Region*). —Region rb is an open container with interior rc.
UprightContainerShape(rb,rc: *Region*)
— Region rb is an upright open container with interior rc.
SimpleUprightContainerShape(rb,rc: *Region*).
OpenContained(ra,rb: *Region*) — Region ra is in the open container rb.
ConvexHull (r: *Region*) → *Region.* — Convex hull of region r.
FullyOutside (ra,rb: *Region*) — Regions ra and rb are separable by a plane
PartiallyContained(ra,rb: *Region*) — Region ra is partially contained in the open container rb.

Definitions:

S.C.D.1 $\text{Cavity}(u,v) \Leftrightarrow$
$\text{IntConn}(u) \wedge \text{IntConn}(v) \wedge \text{DR}(u,v) \wedge$
$\forall_r \text{IntConn}(r) \wedge \text{O}(r,u) \wedge \text{DR}(r,v) \Rightarrow \text{P}(r,u).$

Region u is a cavity of v if it is a maximal interior-connected region disjoint from v. (Note that the outside of v does not satisfy this condition, since u must be a region and by definition a region is bounded.)

S.C.D.2 $\text{Outside}(u,v) \Leftrightarrow [\text{DR}(u,v) \wedge [\forall_w \text{Cavity}(w,v) \Rightarrow \text{DR}(u,w)]].$
Region u is outside v if u is disjoint from v and from every cavity of u.

S.C.D.3 $\text{Contained}(u,v) \Leftrightarrow \exists_c \text{Cavity}(c,v) \wedge \text{P}(u,c).$
Region u is contained in v if u is part of a cavity of v.

S.C.D.4 $\text{CombinedContainer}(ra,rb,rc) \Leftrightarrow$
$\text{EC}(ra,rb) \wedge \text{Cavity}(rc,\text{RUnion}(ra,rb)) \wedge \neg\text{Cavity}(rc,ra) \wedge \neg\text{Cavity}(rc,rb).$

S.C.D.5 $\text{SimpleUprightContainerShape}(rb,rc) \Leftrightarrow$
$\text{UprightContainerShape}(rb,rc) \wedge \forall_{rd} \text{UprightContainerShape}(rb,rd) \Rightarrow \text{P}(rd,rc).$

S.C.D.6 $\text{OpenContained}(ra,rb) \Leftrightarrow \exists_{rc} \text{OpenContainerShape}(rb,rc) \wedge \text{P}(ra,rc).$

S.C.D.7 $\text{FullyOutside}(ra,rb) \Leftrightarrow \text{DC}(\text{ConvexHull}(ra),\text{ConvexHull}(rb))$

S.C.D.8 $\text{PartiallyContained}(ra,rb) \Leftrightarrow$
$\neg\text{OpenContained}(ra,rb) \wedge \exists_{rc}\, P(rc,ra) \wedge \text{OpenContained}(rc,rb).$

Axioms:
S.C.A.1 $\text{Contained}(u,v) \wedge \text{Contained}(v,w) \Rightarrow \text{Contained}(u,w).$
S.C.A.2 $\text{UprightContainerShape}(rb,rc) \Rightarrow \text{OpenContainerShape}(rb,rc).$
S.C.A.3 $\text{OpenContainerShape}(u,v) \Rightarrow \text{EC}(u,v).$
S.C.A.4 $\forall_{r:\,\mathit{Region}}\, P(r,\text{ConvexHull}(r)).$
S.C.A.5 $\text{Cavity}(u,v) \Rightarrow P(u,\text{ConvexHull}(v)).$
S.C.A.6 $\text{OpenContainerShape}(u,v) \Rightarrow P(v,\text{ConvexHull}(u)).$
S.C.A.7 $P(u,v) \Rightarrow P(\text{ConvexHull}(u),\text{ConvexHull}(v)).$
S.C.A.8 $\text{OpenContained}(u,v) \wedge \text{DR}(\text{ConvexHull}(u),v) \Rightarrow$
$\text{OpenContained}(\text{ConvexHull}(u),v).$

7.3.3 Much Smaller

We include a qualitative comparator on the size of regions: MuchSmaller(ra,rb), meaning that region ra is much smaller than rb. This comparator on region is related to the predicate SmallSet(s,r) (section 7.5.6) which in turn is used in some specialized physical axioms (e.g. A.C.A.A, section 7.8.2).

The axioms state that MuchSmaller is a partial ordering (S.M.A.1, .2); compatible with the part relation P (S.M.A.3); and invariant under taking the convex hull (S.M.A.4). S.M.A.5 asserts that a container, closed or open, cannot be much smaller than the region it contains.. It follows that a small region cannot contain a larger region in any sense of "containment". These axioms, and further properties stated below, are satisfied under various possible definitions of MuchSmaller; for example, they are satisfied if MuchSmaller(ra,rb) is defined as the diameter of ra is k times smaller than the diameter of rb for some fixed $k > 1$. (On the other hand, there are plausible geometric relations that could be called "much smaller" that would not satisfy the axioms; for instance, the relation "volume of ra is much smaller than volume of rb" would not satisfy axiom S.M.A.4. or S.M.A.5.).

Symbols:
MuchSmaller(ra,rb: *Region*).

Axioms:

S.M.A.1 $\neg\text{MuchSmaller}(ra,ra).$
S.M.A.2 $\text{MuchSmaller}(ra,rb) \wedge \text{MuchSmaller}(rb,rc) \Rightarrow \text{MuchSmaller}(ra,rc).$
S.M.A.3 $\text{MuchSmaller}(ra,rb) \wedge P(rc,ra) \wedge P(rb,rd) \Rightarrow \text{MuchSmaller}(rc,rd).$
S.M.A.4 $\text{MuchSmaller}(ra,rb) \Rightarrow \text{MuchSmaller}(\text{ConvexHull}(ra),rb)$
S.M.A.5 $\text{MuchSmaller}(ra,rb) \Rightarrow \neg\text{Cavity}(rb,ra) \wedge \neg\text{OpenContainerShape}(ra,rb)$

7.4 Objects

The theory of objects introduces two sorts: *Object* and *ObjectSet*. Objects are disjoint; they do not overlap, and one object is not part of another.

7.4.1 Object Sets

The relations over object sets and their definitions are standard. The sole axiom O.S.A.1 is the axiom of extension, that two sets with the same elements are equal. We do not posit a comprehension axiom, that any definable property defines a set, since that would require an axiom schema. For that reason, when we need to refer to a particular set, we need to posit its existence (in some cases implicitly, by using a constant or function symbol.) For instance, in section 7.4.2 we define the set of all objects in region r at time t to be a set by introducing the function Contents(t,r) to have sort *ObjectSet* and by asserting necessary and sufficient conditions for an object o to be an element of Contents(t,r) (definition O.T.D.2).

Symbols:

Element(x: *Object*; s: *ObjectSet*). — *Object* x is an element of *ObjectSet* s.
Null → *ObjectSet*.
Singleton(x: *Object*) → *ObjectSet*. — { x }
Pair(x,y: *Object*) → *ObjectSet*. — { x,y }
Subset(sa,sb: *ObjectSet*).
Disjoint(sa,sb: *ObjectSet*).
Union(sa,sb: *ObjectSet*) → *ObjectSet*.

Definitions:

O.S.D.1 $\forall_{x}$ ¬Element(x,Null).

O.S.D.2 $\forall_{x,y:Object}$ Element(y,Singleton(x)) ⇔ y=x.

O.S.D.3 $\forall_{sa,sb:\ ObjectSet}$ Subset(sa,sb) ⇔ $\forall_{o}$ Element(o,sa) ⇒ Element(o,sb).

O.S.D.4 $\forall_{x,y,z:\ Object}$ Element(z,Pair(x,y)) ⇔ z=x ∨ z=y.

O.S.D.5 $\forall_{sa,sb:\ ObjectSet}$ Disjoint(sa,sb) ⇔ ¬$\exists_{o}$ Element(o,sa) ∧Element(o,sb).

O.S.D.6 $\forall_{sa,sb:\ ObjectSet;\ x:Object}$ Element(x,Union(sa,sb)) ⇔
Element(x,sa) ∨ Element(x,sb).

Axiom

O.S.A.1 $\forall_{sa,sb:\ ObjectSet}$ [$\forall_{x}$ Element(x,sa) ⇔ Element(x,sb)] ⇒ sa=sb.

7.4.2 Objects and Object Sets: Spatio-Temporal

We next define the primitives that relate objects to the regions they occupy at a given time. The function Place(t,o) is the region the object o occupies at time t. The predicate FeasiblePlace(o,r) holds if it is physically possible to configure o so that it occupies r. The predicate OSPlace(t,s,r) holds if r is the region occupied by object set s at time t. (This is a predicate rather than a function, since the null set does not occupy any region.) The function Contents(t,r) is the set of objects that are in region r at time t.

Symbols:

Place(t: *Time*; o: *Object*) → *Region.*
FeasiblePlace(o: *Object*; r: *Region*).
OSPlace(t: *Time*; s: *ObjectSet*; r: *Region*).
Contents(t: *Time*; r: *Region*) → *ObjectSet.*

Definitions:

O.T.D.1 $\mathrm{OSPlace}(t,s,r) \Leftrightarrow$

$[\forall_{o}\ \mathrm{Element}(o,s) \Rightarrow \mathrm{P}(\mathrm{Place}(t,o),r)] \land$

$[\forall_{ra}\ [\forall_{o}\ \mathrm{Element}(o,s) \Rightarrow \mathrm{P}(\mathrm{Place}(t,o),ra)] \Rightarrow \mathrm{P}(r,ra)].$

The region occupied by a set s is the minimal region that contains all the regions occupied by the elements of s.

O.T.D.2 $\forall_{o:\ \mathit{Object};\ r:\ \mathit{Region};\ t:\ \mathit{Time}}\ \mathrm{Element}(o,\mathrm{Contents}(t,r)) \Leftrightarrow \mathrm{P}(\mathrm{Place}(t,o),r).$

O.T.D.3 $\forall_{t:\ \mathit{Time};\ o:\ \mathit{Object}}\ \mathrm{FeasiblePlace}(o,r) \Leftrightarrow \exists_{t:\ \mathit{Time}}\ \mathrm{Place}(t,o)=r.$

Axioms:

O.T.A.1 $\forall_{p,q:\ \mathit{Object};\ t:\ \mathit{Time}}\ p \neq q \Rightarrow \mathrm{DR}(\mathrm{Place}(t,p), \mathrm{Place}(t,q)).$

Any two objects are spatially disjoint.

O.T.A.2 $\neg\exists_{t,r}\ \mathrm{OSPlace}(t,\mathrm{Null},r).$

The null set has no place.

O.T.A.3 $\forall_{s:\ \mathit{ObjectSet};\ t:\mathit{Time}}\ s \neq \mathrm{Null} \Rightarrow \exists^{1}_{r}\ \mathrm{OSPlace}(t,s,r).$

Every non-empty set of objects occupies a unique region at any time.

O.T.A.4 $\mathrm{FeasiblePlace}(o,r) \Rightarrow \mathrm{IntConn}(r).$

An object occupies an interior connected region.

O.T.A.5 $\mathrm{OSPlace}(t,s,r) \land \mathrm{Object}(o) \land \neg\mathrm{Element}(o,s) \Rightarrow \mathrm{DR}(\mathrm{Place}(t,o),r)$

Any object o that is not an element of set s occupies a region disjoint from the place of s. You would think this should be a consequence of O.T.A.5 and O.T.A.2, but a more powerful spatial theory would be needed to support that inference.

7.4.3 Objects containing regions

We here define the containment relations between a container, which is an object or a set of objects and a region that it contains. Here and in section 7.5.4, we define closed containers in terms of a set of objects but open containers in terms of a single object, because closed containers are often composed of multiple objects (e.g. a box with a lid; a bottle with a cap; and so on) whereas this is rarer for open containers, though it does occur (e.g. cupping your two hands.)

Note: A cup upside down inside a closed box is both an object inside a closed container and part of a closed container. A box with shelves therefore forms *n(n-1)/2* closed containers (any pair of shelves/top/bottom determine a container) and a box with small cubby holes and dividers in two directions forms an exponential number (any interior-connected collection of cubby holes is considered a closed container) but that's the way it goes.

Symbols:
ClosedContainer(t: *Time*; s: *ObjectSet;* rc: *Region*).
OpenContainer(t: *Time;* o: *Object;* rc: *Region*).
UprightContainer(t: *Time;* o: *Object;* rc: *Region*).
SimpleUprightContainer(t: *Time;* o: *Object;* rc: *Region*).

Definitions:

O.R.D.1 ClosedContainer(t,s,rc) $\Leftrightarrow \exists_{rs}$ OSPlace(t,s,rs) $\wedge$ Cavity(rc,rs).

O.R.D.2 OpenContainer(t,o,rc) $\Leftrightarrow$
Time(t) $\wedge$ Object(o) $\wedge$ OpenContainerShape(Place(t,o),rc).

O.R.D.3 UprightContainer(t,o,rc) $\Leftrightarrow$
Time(t) $\wedge$ Object(o) $\wedge$ UprightContainerShape(Place(t,o),rc).

O.R.D.4 SimpleUprightContainer(t,o,rc) $\Leftrightarrow$
Time(t) $\wedge$ Object(o) $\wedge$ SimpleUprightContainerShape(Place(t,o),rc).

7.4.4 Object Containment

We define the analogous containment relations for the case of one object or a set of objects containing another object.

Symbols:
CContained(t: *Time*; ox: *Object;* s: *ObjectSet*).
OContained(t: *Time*; ox, ob: *Object*).
UContained(t: *Time*; ox, ob: *Object*).
CContents(t: *Time;* s: *ObjectSet)* → *ObjectSet*.
UContents(t: *Time;* o: *Object)* → *ObjectSet*.

Definitions:

O.C.D.1 CContained(t,ox,s) $\Leftrightarrow$
$\exists_{rc}$ ClosedContainer(t,s,rc) $\wedge$ Object(ox) $\wedge$ P(Place(t,ox),rc).

O.C.D.2 OContained(t,ox,ob) $\Leftrightarrow$
$\exists_{rc}$ OpenContainer(t,ob,rc) $\wedge$ Object(ox) $\wedge$ P(Place(t,ox),rc).

O.C.D.3 UContained(t,ox,ob) $\Leftrightarrow$
$\exists_{rc}$ UprightContainer(t,ob,rc) $\wedge$ Object(ox) $\wedge$ P(Place(t,ox),rc).

O.C.D.4 $\forall$ t: *Time;* s,sb: *ObjectSet* s = CContents(t,s,sb) $\Leftrightarrow$
$\forall_{ox}$ Element(ox,s) $\Leftrightarrow$ CContained(t,ox,sb).

O.C.D.5 $\forall$ t: *Time;* ob: *Object;* s: *ObjectSet* s = UContents(t,ob) $\Leftrightarrow$
$\forall_{ox}$ Element(ox,s) $\Leftrightarrow$ UContained(t,ox,ob).

7.4.5 Fits and Small Set

The final category of spatial relations that we axiomatize involves objects fitting into a region. The predicates Fits(s,r) means that object set s fits into region r. (This is a purely geometric relation, meaning that there is a configuration of s that lies inside r; it does not require that it is physically possible to move the objects in s into that configuation.) The predicate SmallSet(s,r) means that object set s fits into a region that is much smaller than r. Likewise we consider a class of *small objects*. These are objects that are much smaller than the agent, and therefore particularly easy to move.

Symbols:
Fits(s: *ObjectSet;* r: *Region*).
OMuchSmaller(o: *Object;* r: *Region*).
SmallSet(s: *ObjectSet;* r: *Region*).
SmallObject(o: *Object*).

Definition:

O.F.D.1 $\mathrm{Fits}(s,r) \Leftrightarrow s=\mathrm{Null} \lor \exists_{t,ra}\,\mathrm{OSPlace}(t,s,ra) \land P(ra,r)$

O.F.D.2 $\mathrm{OMuchSmaller}(o,r) \Leftrightarrow$

$$\forall_{rb}\ \mathrm{FeasiblePlace}(o,rb) \Rightarrow \mathrm{MuchSmaller}(rb,r).$$

O.F.D.3 $\mathrm{SmallObject}(o) \Leftrightarrow$

$$\forall_{ra}\ \mathrm{FeasiblePlace}(\mathrm{Agent},ra) \Rightarrow \mathrm{OMuchSmaller}(o,ra).$$

O.F.D.4 $\mathrm{SmallSet}(s,r) \Leftrightarrow$

$$[\exists_{ra}\ \mathrm{Fits}(s,ra) \land \mathrm{MuchSmaller}(ra,r)] \land$$
$$[\forall_{o}\ \mathrm{Element}(o,s) \Rightarrow \mathrm{OMuchSmaller}(o,ra)]$$

7.4.6 Isolates

The relation Isolates(t,sp,sx), read “At time t, object set sp isolates object set sx,” is central to our formulation of frame axioms in section 7.5.4. The intended meaning is that the agent, in his current position, cannot move the objects in sx without moving some of the objects in sp. We do not give a full characterization of the predicate Isolates; we enumerate a few properties and give two sufficient conditions. Definition O.I.D.1 defines sets sp and sx as “isolate candidates” if they are disjoint, and neither contains the agent himself. Axiom O.I.A.1 states that sp can only isolate sx if they are isolate candidates. Axiom O.I.A.2 states that sp isolates sx if they are isolate candidates, and if the only objects in contact with an object in sx is either in sx itself or in sp. Axiom O.I.A.3 states that sp isolates sx if sp forms a closed container with cavity rc, sx is the contents of rc, and the agent is outside rc.

One might suppose that O.I.A.3 followed from O.I.A.1 and .2, but figure 13 shows why that is not the case. The agent is in contact with the object U, so the conditions of axiom O.I.A.2 are not satisfied, but axiom O.I.A.3 can be used to infer that the set { V } isolates set { U }.

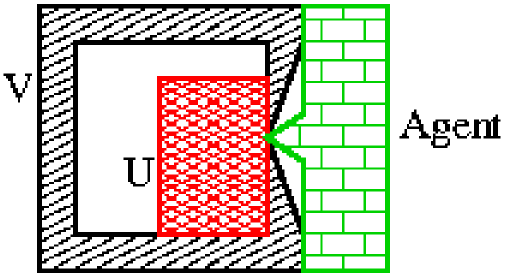


Figure 11: Axiom O.I.A.3

Symbol:

Isolates(t: *Time*; sp,sx: *ObjectSet*)
IsolatesCand(sp,sx: *ObjectSet*)

Definition:
O.I.D.1 IsolateCand(sp,sx) ⇔
$\neg$Element(Agent,sp) $\wedge$ $\neg$Element(Agent,sx) $\wedge$ Disjoint(sp,sx).

Axioms:

O.I.A.1 Isolates(t,sp,sx) ⇒ IsolateCand(sp,sx)

O.I.A.2 $\forall_{sx,sp:\ ObjectSet;;\ t:\ Time;}$ IsolateCand(sp,sx) $\wedge$
[$\forall_{ox,o}$ Element(ox,sx) $\wedge$ EC(Place(t,o),Place(t,ox)) ⇒
[Element(o,sx) $\vee$ Element(o,sp)]] ⇒
Isolates(t,sp,sx).

O.I.A.3 ClosedContainer(t,sp,rc) $\wedge$ $\neg$P(Place(t,Agent),rc) $\wedge$ $\neg$Element(Agent,sp) ⇒
Isolates(t,sp,Contents(t,rc))

7.5 Motion and Manipulation

In this section we present a partial, qualitative theory of an object moving and of an agent manipulating an object. Our theory partially categorizes motion under quasi-static conditions ((LaValle, 2006), section 13.1.3), in which forces like friction always quickly dissipate inertia.

Our theory posits the following constraints on motion:

1. The agent can move as he chooses, subject to the limits on the possible configuration he can attain.
2. The agent can grasp an object and manipulate it.
3. An object that is in an unstable position will fall until it attains a stable position.

4. An object that is specified to be fixed remains motionless. (Boundary conditions in problem specifications often involve objects that are assumed to be fixed.)
5. If a collection of objects are in a stable position, are not grasped, and are isolated from any moving object other than the agent, then they remain motionless.

Otherwise, the theory is indeterminate about the motion of the object. Our theory does not specify any constraints on the trajectory of a falling object, except for a rule that states that an object that is inside an upright open container cannot fall out of the container (axiom H.U.A.1, section 7.6.3).

7.5.1 Grasping an Object

Our theory of grasping an object has two basic primitives. The constant Agent denotes the hero agent, a distinguished object. The predicate Grasp(t,o) meaning that the agent is grasping object o at time t. We define some further predicates for convenient reference.

By convention we suppose that, on any time line, the agent grasps any given object over a time interval that is open on the left and closed on the right. Thus, if the agent grasps o from time ta to tb and then releases it, he is grasping o at time tb and is not grasping it over some interval (tb,tc] open on the left and closed on the right. For instance, in the branching structure shown in figure 12 on time line 1, the agent grasps o from ta to tc; on time line 2, the agent grasps o from ta to tb and then is not grasping at all times after tb up to td.

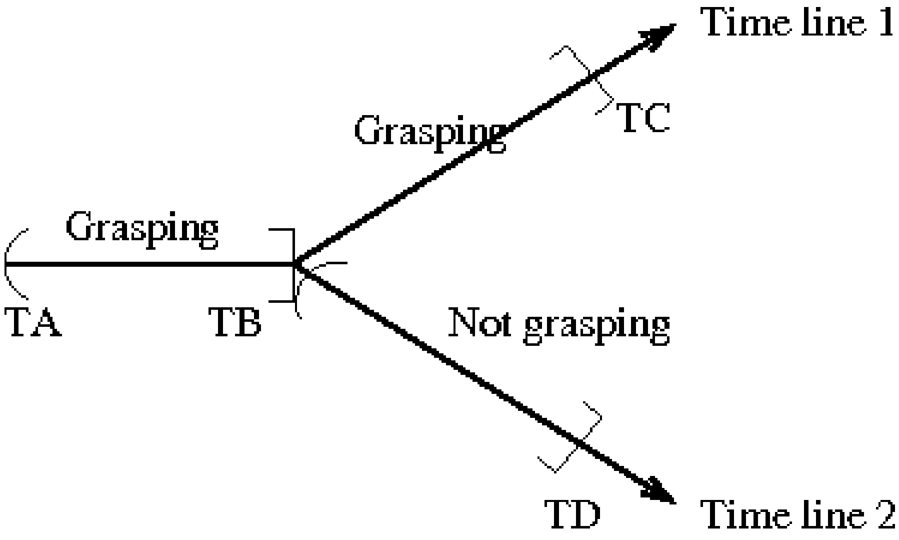


Figure 12: Grasping on a dense branching time line

Symbols:
Agent → *Object.*
Grasp(t: *Time*; o: *Object*).
EmptyHanded(t: *Time*).
Grasps(ta,tb: *Time*; o: *Object:).*
CanGrasp(t: *Time*; o: *Object*). — The agent can grasp object o at time t.
Released(ta,tb: *Time*; o: *Object*).

Definitions:

M.G.D.1 $\text{EmptyHanded}(t) \Leftrightarrow \text{Time}(t) \wedge \neg\exists_{o}\, \text{Grasp}(t,o)$.
The agent is EmptyHanded at time t if he is not grasping anything at t.

M.G.D.2 $\text{Grasps}(ta,tb,o) \Leftrightarrow \text{Lt}(ta,tb) \wedge \forall_{t}\, \text{Lt}(ta,t) \wedge \text{Leq}(t,tb) \Rightarrow \text{Grasp}(t,o)$.
The agent grasps object o from time ta (non-inclusive) to time tb (inclusive).

M.G.D.3 $\text{CanGrasp}(t,o) \Leftrightarrow \exists_{tb}\, \text{Grasps}(t,tb,o)$.

M.G.D.4 $\text{Released}(ta,tb,o) \Leftrightarrow$
$\text{Object}(o) \wedge \text{Lt}(ta,tb) \wedge \forall_{t}\, \text{Lt}(ta,t) \wedge \text{Leq}(t,tb) \Rightarrow \neg\text{Grasp}(t,o)$.
Object o is released (i.e. the agent is not grasping it) from time ta (not inclusive) through tb (inclusive).

Axioms:

M.G.A.1 $\text{Isolates}(t,sp,sx) \wedge \text{Element}(o,sx) \Rightarrow \neg\text{Grasp}(t,o) \wedge \neg\text{CanGrasp}(t,o)$.
The agent cannot grasp an object o that is part of an isolated set sx.

M.G.A.2 $\forall_{ta:\, \mathit{Time};\; o:\, \mathit{Object}}\, \exists_{tb}\, \text{Released}(ta,tb,o)$.
At any time ta it is possible for the agent to release object o (assuming that he is holding o).

M.G.A.3 $[\text{Grasp}(t,o) \vee \text{CanGrasp}(t,o)] \Rightarrow$
$\neg\text{CContained}(t,\text{Agent},\text{Singleton}(o)) \wedge \neg\text{OContained}(t,\text{Agent},o)$.
The agent cannot grasp a container (in order to move it) if he is entirely inside it.

M.G.A.4 $[\text{Grasp}(t,o) \vee \text{CanGrasp}(t,o)] \Rightarrow \text{EC}(\text{Place}(t,\text{Agent}),\text{Place}(t,o))$.
The agent can only grasp an object if he is spatially touching it.

7.5.2 Motion

We introduce some convenient symbols for describing motion and manipulation. We define all of these in terms of change of place and grasping except the predicate Moving(t,o). The predicate Moving(t,o) is implicitly defined in axiom M.O.A.1, which states that object o is Motionless between times ta and tb if and only if it is not Moving at any time t between ta and tb.

Symbols:

Moves(ta,tb: *Time;* o: *Object).*
Object o changes place from time ta to time tb.

Moving(t: *Time;* o: *Object).*
Object o is moving at time t.

Motionless(ta,tb: *Time;* o: *Object).*
Object o is motionless between time ta and time tb.

TravelTo(r: *Region*) → *Action*.
The action of the agent traveling empty-handed to region r.

StandStill → *Action.*
The action of the agent standing still, not changing his grasps..

.

Definitions:

M.O.D.1 $\forall_{\mathrm{ta,tb}:\mathit{Time};\ \mathrm{o}:\mathit{Object}}\ \mathrm{Moves(ta,tb,o)} \Leftrightarrow \mathrm{Lt(ta,tb)} \wedge \mathrm{Place(tb,o)} \neq \mathrm{Place(ta,o)}.$

M.O.D.2 1 $\forall_{\mathrm{ta,tb}:\mathit{Time};\ \mathrm{o}:\mathit{Object}}\ \mathrm{Motionless(ta,tb,o)} \Leftrightarrow$
$\mathrm{Lt(ta,tb)} \wedge \forall_{\mathrm{t}:\mathit{Time}}\ \mathrm{Leq3(ta,t,tb)} \Rightarrow \mathrm{Place(t,o)} = \mathrm{Place(ta,o)}.$

M.O.D.3 1 $\forall_{\mathrm{ta,tb}:\mathit{Time};\ \mathrm{r}:\mathit{Region}}\ \mathrm{Occurs(ta,tb,TravelTo(r))} \Leftrightarrow$
$\mathrm{r} = \mathrm{Place(tb,Agent)} \wedge \forall_{\mathrm{o}:\mathit{Object}}\ \mathrm{Released(ta,tb,o)}.$

M.O.D.4 $\mathrm{Occurs(ta,tb,StandStill)} \Leftrightarrow$
$\mathrm{Motionless(ta,tb,Agent)} \wedge$
$\forall_{\mathrm{t,o}}\ \mathrm{Leq3(ta,t,tb)} \Rightarrow [\mathrm{Grasp(t,o)} \Leftrightarrow \mathrm{Grasp(ta,o)}]$

Axiom:

M.O.A.1 $\mathrm{Motionless(ta,tb,o)} \Leftrightarrow \mathrm{Lt(ta,tb)} \wedge [\forall_{\mathrm{t}:\mathit{Time}}\ \mathrm{Lt(ta,t)} \wedge \mathrm{Lt(t,tb)} \Rightarrow \neg\mathrm{Moving(t,o)}].$

7.5.3 Stability and Falling

We next present a very partial theory of stability and falling. We assume a world in which, normally, everything is in a stable state or is being grasped; this is called an AllStable state of the world (definition M.S.D.1). This happy condition can be interrupted if the agent drops object o, which he does by ungrasping it in an unstable position. That will cause o to fall, which may in turn destabilize other objects, causing them to fall. However, if the agent stands still, then the world will eventually attain a state where everything is stable (axiom M.S.A.1; see further discussion below).

We do not give any geometric or physical conditions for stability, either necessary or sufficient; nor do we give any constraints on the motions of falling objects, except to posit that falling objects inside containers do not exit the container nor cause anything outside the container to fall (axiom M.R.A.5, section 7.5.4). Further constraints on how avalanches of falling objects spread are given as frame axioms in section 7.5.4.

An object can be declared to be Fixed, in which case it is always stable and motionless (M.S.A.2). This is particularly useful in problem specifications; there are often fixed objects such as the ground, tables, buildings, and so on. It is taken to be an atemporal (eternal) property of the object. The predicate AllStable holds at time t if all objects in the world are either stable or being grasped. The set AllMobileObjects includes every mobile object i.e. not fixed and not the agent.

Axiom M.S.A.1 is not as strong as one would wish. The axiom states that, starting at any time ta, there is a timeline in which the agent stands still and in which the world eventually reaches an AllStable state. What one would like to say, rather, is that, whatever the agent does, as long has he doesn't keep dropping objects, the world will eventually reach an AllStable state. However, stating this would require a significantly more expressive language of time, that allows quantification over time-lines. The formulation here is sufficient for the inferences we are considering.

Symbols:

Stable(t: *Time;* o: *Object*) —
At time t, object o is in a position where it will be stable, if released.
AllStable(t: *Time).* — All objects are either grasped or stable at time t.
Fixed(o: *Object)* — Object o is fixed in place.
AllMobileObjects → *ObjectSet* — The set of all non-fixed objects.

Definitions:

M.S.D.1 $\text{AllStable}(t) \Leftrightarrow \forall_{o:Object}\ o = \text{Agent} \vee \text{Stable}(t,o) \vee \text{Grasp}(t,o).$

M.S.D.2 $\forall_{o:Object}\ \text{Element}(o,\text{AllMobileObjects}) \Leftrightarrow o \neq \text{Agent} \wedge \neg\text{Fixed}(o).$

Axioms:

M.S.A.1 $\forall_{ta:Time}\ \exists_{tb}\ \text{Occurs}(ta,tb,\text{StandStill}) \wedge \text{AllStable}(tb).$

M.S.A.2 $\text{Fixed}(o) \wedge \text{Lt}(ta,tb) \wedge \text{Time}(t) \Rightarrow \text{Motionless}(ta,tb,o) \wedge \text{Stable}(t,o)$

A fixed object is stable and motionless.

7.5.4 Frame axioms

In this section we present frame axioms, which limit the way that change over time can occur; that is, they specify conditions under which things remain the same.

The first axiom M.R.A.1 asserts that an object o is moving at time t only if o is the agent, or is not stable at t or some object ox is being manipulated at time t (that is, ox is being grasped and is moving.) This, in itself, has the form of a state constraint, rather than a frame axiom. However, in combination with axiom M.O.A.1, which asserts that an object which is never moving remains motionless, and definition M.O.D.2, which asserts that a motionless object remains in the same place, the net effect is to posit that an object o can change its position between times ta and tb only if o is the agent, if o is unstable at some time between ta and tb, or if some (possibly other) object ox is being manipulated between ta and tb.

Axiom M.R.A.2 is based on the idea of a causally isolated set of objects. A set of objects sx is *causally isolated* between times ta and tb if there is a set of objects sp that isolates sx and that remains motionless throughout the interval [ta,tb]. For instance, if a container remains motionless over a time interval and the agent remains outside, then the set of objects inside is causally isolated. A set s is *static causally isolated* if it is causally isolated and, additionally, all the objects in s are stable at the initial time ta (definition M.R.D.2). Frame axiom M.R.A.2 asserts that if set s is static causally isolated, then every object in s remains motionless and remains stable.

Frame axioms M.R.A.3 and M.R.A.4 limit the influence of objects moving around inside a container on objects outside the container. They state that, for any container, if all the objects not inside the container (including the container itself) are stable at time ta and are not manipulated over the interval [ta,tb], then all these objects remain stable and motionless throughout [ta,tb]. M.R.A.3 states this for a set of objects s that forms a closed container. M.R.A.4 states it for an object o that is an upright container; in that case, it is necessary to add the condition that there are no objects partially inside the container except possibly the agent.

In M.R.A.2 – .4 one would prefer to make the stronger statement that in general if two sets of objects are causally isolated one from another, then they evolve independently. The axioms here and the analogous axiom A.S.A.2 below (section 7.8.1) are essentially the special case of this principle in the case where one of these evolutions is that all the objects remain stable and motionless. Formulating the more general principle requires a more general notion of a history than we develop here and a powerful calculus on histories (Davis, 2011).

Finally, axioms M.R.A.5 asserts that, if everything is stable at time ta, and the agent does not grasp anything between times ta and tb, then everything remains stable and motionless from ta to tb.

Symbols:
CausallyIsolated(ta, tb : *Time*; s: *ObjectSet*).
StaticCausallyIsolated(ta, tb : *Time*; s: *ObjectSet*).
StableThroughout (ta, tb : *Time*; o: *Object*).
NoPartialContents(t: *Time;* o: *Object)* —

No objects except possibly the agent are partially contained in the open container o at time t.

Definitions:

M.R.D.1 CausallyIsolated(ta, tb,sx) $\Leftrightarrow$
Lt(ta,tb) $\wedge$
$\exists_{sp:ObjectSet}$ [$\forall_o$ Element(o,sp) $\Rightarrow$ Motionless(ta,tb,o)] $\wedge$
[$\forall_t$ Leq(ta,t) $\wedge$ Lt(t,tb) $\Rightarrow$ Isolates(t,sp,sx)].

M.R.D.2 StaticCausallyIsolated(ta, tb,s) $\Leftrightarrow$
CausallyIsolated(ta, tb,s) $\wedge$ $\forall_{ox}$ Element(ox,s) $\Rightarrow$ Stable(ta,ox)

M.R.D.3 StableThroughout(ta, tb,o) $\Leftrightarrow$
Lt(ta,tb) $\wedge$ [$\forall_t$ Leq3(ta,t,tb) $\Rightarrow$ Stable(t,o)].

M.R.D.4 $\forall_{t:\,Time;\;o:\,Object}$ NoPartialContents(t,o) $\Leftrightarrow$
$\forall_{oc:\,Object}$ PartiallyContained(Place(t,oc),Place(t,o)) $\Rightarrow$
oc = Agent.

Axioms:

M.R.A.1 Moving(t,o) $\Rightarrow$
o = Agent $\vee$ $\exists_{ox}$ $\neg$Stable(t,ox) $\vee$ (Grasp(t,ox) Moving(t)).

M.R.A.2 StaticCausallyIsolated(ta,tb,s) $\wedge$ Element(o,s) $\Rightarrow$
Motionless(ta,tb,o) $\wedge$ StableThroughout(ta,tb,o).

M.R.A.3 $\forall_{ta,tb:\,Time;s:ObjectSet}$
[$\forall_{ox:\,Object}$ ox =Agent $\vee$ CContained(ta, ox,s) $\vee$
[Stable(ta,ox) $\wedge$ Released(ta,tb,ox)]]

$\Rightarrow$
$[\forall_{ox:\ Object}\ ox = Agent \lor CContained(ta,ox,s) \lor$
$[Motionless(ta,tb,ox) \land StableThroughout(ta,tb,ox)]].$

M.R.A.4 $\forall_{ta,tb:\ Time; ob: Object}\ NoPartialContents(ta,ob) \land$
$[\forall_{ox:\ Object}\ ox = Agent \lor UContained(ta,ox,ob) \lor$
$[Stable(ta,ox) \land Released(ta,tb,ox)]]$
$\Rightarrow$
$[\forall_{ox:\ Object}\ ox = Agent \lor UContained(ta,ox,ob) \lor$
$[Motionless(ta,tb,ox) \land StableThroughout(ta,tb,ox)]].$

M.R.A.5 $Lt(ta,tb) \land AllStable(ta) \land EmptyHanded(ta) \land$
$[\forall_{o:\ Object}\ Released(ta,tb,o)] \Rightarrow$
$Motionless(ta,tb,o) \land AllStable(tb)$

7.5.5 Feasibility of Travelling

We give a partial characterization of the feasibility of TravelTo (i.e. movements of the agent while empty-handed).

The predicate Trajectory(ra,rb,rw) means that there is a feasible trajectory for the agent from ra to rb remaining in rw, assuming that rw is free of obstacles. We give some necessary conditions for this (axiom M.F.A.2) and some combinatorial axioms (M.F.A.4-.6). Axiom M.F.A.3 covers the trivial case where the agent stays fixed in ra.

Axioms M.F.A.7 gives necessary conditions and M.F.A.8 give sufficient conditions for the feasibility of travelling in terms of Trajectory if no other objects are moving.[11] M.F.A.7 states that, if TravelTo(rb) occurs from ta to tb, then there exists a region rw such that Trajectory(Place(ta,Agent),rb,rw) and the agent stays in rw during [ta,tb]. M.F.A.8 states that, if Trajectory(Place(ta,Agent),rb,rw) and rw is free of obstacles, thenTravelTo(rb) is feasible at time ta.

The predicate Graspable(t,o), meaning that the agent can travel to a position where he can grasp o, defined in definition M.F.D.4, is not used in the axioms, but is used in the problem specification for scenario 4.

Symbols:
NoObstacles(t: *Time;* r: *Region*) —
No objects other than the agent are inside region r at time t.
Trajectory(ra,rb,rw: *Region*). Discussed in the text.
MiddlePos(ta,tb: *Time*: o: *Object;* r: *Region*) —
Object o occupies region r some time between times ta and tb.
StaysIn(ta,tb: *Time;* o: *Object;* r: *Region*) —
Object o remains inside r throughout the interval [ta,tb].
Graspable(t: *Time*; o: *Object*) — At time t, the agent can move so as to grasp o.

[11] If other objects are falling, then it would be difficult to give either necessary or sufficient conditions, since an external object may either fall so as to block the path or fall so as to clear the path.

Definitions:

M.F.D.1 $\text{NoObstacles}(t,r) \Leftrightarrow$

$\text{Time}(t) \land \forall_{o:Object}\ o \neq \text{Agent} \Rightarrow \text{DR}(\text{Place}(t,o),r).\ _{o:Object}$

M.F.D.2 $\text{MiddlePos}(ta,tb,o,r) \Leftrightarrow$

$\text{Object}(o) \land \exists_{tx}\ \text{Leq3}(ta,tx,tb) \land\ r=\text{Place}(tx,o).$

M.F.D.3 $\forall_{ta,tb:\ Time;\ o:Object;\ r:Region}\ \text{StaysIn}(ta,tb,o,r) \Leftrightarrow$

$\forall_{rx}\ \text{MiddlePos}(ta,tb,o,rx) \Rightarrow P(rx,r).$

M.F.D.4 $\text{Graspable}(t,o) \Leftrightarrow$

$\exists_{tb,ra}\ \text{Occurs}(t,tb,\text{TravelTo}(ra)) \land \text{CanGrasp}(tb,o).$

Object o is graspable if the agent can travel to a place ra where he can grasp o.

Axioms:

M.F.A.1 $\text{Lt}(ta,tb) \land \text{Object}(o) \Rightarrow \exists_{rw}\ \text{StaysIn}(ta,tb,o,rw).$

In a more powerful spatio-temporal theory this would of course be a theorem rather than an axiom.

M.F.A.2 $\text{Trajectory}(ra,rb,rw) \Rightarrow$

$\text{FeasiblePlace}(\text{Agent},ra) \land \text{FeasiblePlace}(\text{Agent},rb) \land$

$\text{IntConn}(rw) \land P(ra,rw) \land P(rb,rw).$

M.F.A.3 $\text{FeasiblePlace}(\text{Agent},ra) \Rightarrow \text{Trajectory}(ra,ra,ra).$

M.F.A.4 $\text{Trajectory}(ra,rb,rw) \Rightarrow \text{Trajectory}(rb,ra,rw).$

M.F.A.5 $\text{Trajectory}(ra,rb,rw) \land \text{Trajectory}(rb,rc,rx) \Rightarrow$

$\text{Trajectory}(ra,rc,\text{RUnion}(rw,rx)).$

M.F.A.6 $\text{Trajectory}(ra,rb,rw) \land P(rw,rx) \land \text{IntConn}(rx) \Rightarrow$

$\text{Trajectory}(ra,rb,rx).$

M.F.A.7 $\text{EmptyHanded}(ta) \land \text{AllStable}(ta) \land \text{Occurs}(ta,tb,\text{TravelTo}(rb)) \Rightarrow$

$\exists_{rw}\ \text{Trajectory}(\text{Place}(ta,\text{Agent}),rb,rw) \land \text{StaysIn}(ta,tb,\text{Agent},rw) \land$

$\text{NoObstacles}(ta,rw).$

M.F.A.8 $\text{EmptyHanded}(t) \land \text{AllStable}(t) \land \text{NoObstacles}(t,rw) \land$

$\text{Trajectory}(\text{Place}(t,\text{Agent}),rb,rw) \Rightarrow$

$\exists_{tb}\ \text{Occurs}(t,tb,\text{TravelTo}(rb)) \land \text{StaysIn}(t,tb,\text{Agent},rw).$

7.6 Histories

There are some physical constraints whose representation requires the use of *histories*: functions from time to regions (Hayes, 1979). For example, in a continuous model of time, the statement that objects move continuously must be stated as a property of the trajectory of the object over time (axioms H.I.A.1 below).

In a general theory of manipulation, it would be necessary to characterize a large class of manipulations that are physically possible; e.g. that the agent can move its hand through any continuous trajectory of geometrically feasible positions, subjects to constraints of continuity and bounds on the velocity and acceleration. In our framework, such a statement is couched in terms of sufficient condition for the existence of certain kinds of histories; it can be formalized, either by providing a suitably rich constructive vocabulary of manipulation, or by asserting a powerful comprehension axiom, either a second-order axiom or an axiom schema. Examples of how such axioms are formulated and used can be found in (Davis, 2008) and (Davis, 2011). However, these create an immense explosion of the search space in inference. Instead we have a number of specialized axioms and function symbols that guarantee the existence of various histories. For instance, axiom H.I.A.3 guarantees the existence of a constant history for each region.

7.6.1 Basic properties of histories

The basic spatio-temporal primitive associated with histories is the function Slice(t,h), the region that is the extent of history h at time t. The basic primitive relating histories to objects is the function HPlace(o), the history corresponding to the regions occupied by object o.

The predicate Continuous(h) means that history h is continuous with respect to the dual-Hausdorff metric (Davis, 2001). Formalizing this definition would be both lengthy and unnecessary here; we take this to be a primitive.

A history h is *weakly continuous* if it never jumps from one region to a disconnected region. Intuitively, h is *weakly continuous* if a small marble that can predict in advance how h will develop can succeed in staying inside h. Formally, h is weakly continuous at time tm if there is an open interval (tc,td) containing t and a region r such that, for any time t in (tc,td), the slice of h at t contains r (H.I.D.4). The dynamic cavities shown in figure 6 are weakly continuous, though not continuous.

The remaining symbols and the axioms are straightforward.

Symbols:
Slice(t: *Time,* h: *History*) → *Region*. — The slice of history h at time t (a region).
Continuous(ta,tb: *Time*; h: *History*) — History h is continuous between times ta and tb.
HPlace(o: *Object*) → *History*. — The place occupied by object o over time (a history).
HSPlace(s: *ObjectSet*; h: *History*) —
Object set s occupies history h over time. (Like OSPlace, this has to be a relation rather than a function because of the case s=Null).
WeaklyContinuous(ta,tb: *Time*; h: *History*) — Defined in the text.
Constant(t1,t2: *Time*; h: *History*) —
History h has a constant value between times t1 and t2 (inclusive).
HUnion(ha,hb: *History*) →*History*. Spatial union of histories ha and hb
AlwaysIntConn(t1,t2: *Time*; h: *History*) —
History h is always interior-connected between times t1 and t2 (inclusive).

Definitions:

H.I.D.1 AlwaysIntConn(t1,t2,h) ⇔

History(h) ∧ Lt(t1,t2) ∧ $\forall_{t}$ Leq3(t1,t,t2) ⇒ IntConn(Slice(t,h)).

H.I.D.2 $\forall_{s:ObjectSet;\ h:History}$ HSPlace(s,h) ⇔$\forall_{t:Time}$ OSPlace(t,s,Slice(t,h)).

H.I.D.3 Constant(t1,t2,h) ⇔
History(h) ∧ Lt(t1,t2) ∧ $\forall_{t}$ Leq3(t1,t,t2) ⇒ Slice(t,h) = Slice(t1,h).

H.I.D.4 WeaklyContinuous(ta,tb,h) ⇔
Lt(ta,tb) ∧ History(h) ∧ AlwaysIntConn(ta,tb,h) ∧
$\forall_{tm}$ Lt(ta,tm) ∧ Lt(tm,tb) ⇒
$\exists_{tc,td,r}$ Lt(tc,tm) ∧ Lt(tm,td) ∧ $\forall_{t}$ Leq3(tc,t,td) ⇒ P(r,Slice(t,h)).

Axioms:
H.I.A.1 Object(o) ∧ Lt(ta,tb) ⇒ Continuous(ta,tb,HPlace(o)).
An object occupies a continuous history.

H.I.A.2 $\forall_{t:Time;\ o:Object}$ Place(t,o) = Slice(t,HPlace(o)).
Place can be defined in terms of Slice and HPlace.

H.I.A.3 Region(r) ∧ Lt(t1,t2) ⇒ $\exists_{h}$ Constant(t1,t2,h) ∧ Slice(t1,h)=r.
For any region r there is a history that is constantly equal to r.

H.I.A.4 Constant(ta,tb,h) ⇒ Continuous(ta,tb,h).
A constant history is continuous.

H.I.A.5 Continuous(ta,tb,h) ⇒ WeaklyContinuous(ta,tb,h).
A continuous history is weakly continuous.

H.I.A.6 Continuous(ta,tb,h) ∧ Leq(ta,tc) ∧ Lt(tc,td) ∧ Leq(td,tb) ⇒
Continuous(tc,td,h).
A history that is continuous over [ta,tb] is continuous over any subinterval.

H.I.A.7 Continuous(ta,tb,h) ∧ Continuous(tb,tc,h) ⇒ Continuous(ta,tc,h).
A history that is continuous over two adjoining intervals is continuous over their join.

H.I.A.8 $\forall_{t:Time;\ ha,hb:History}$ Slice(t,HUnion(ha,hb)) = RUnion(Slice(t,ha),Slice(t,hb)).

7.6.2 Dynamic containers and cavities

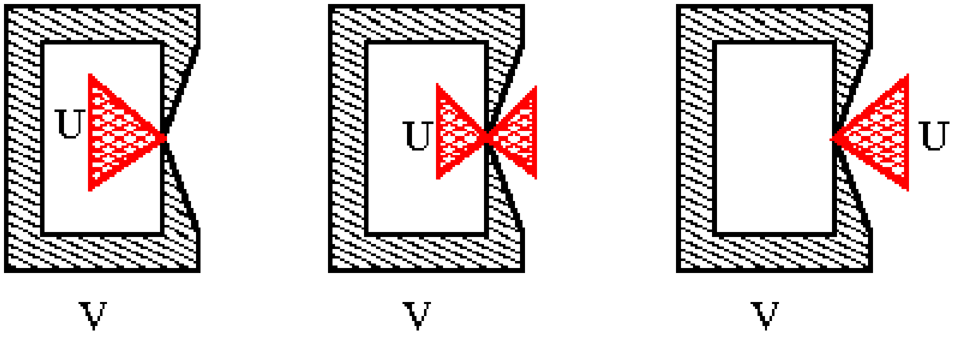
In a container made of flexible material, cavities can split and merge, like bubbles in liquid; they can open up to the outside world, or close themselves off from the outside world.

A history hc is a *dynamic cavity* of container ho over interval [ta,tb] if hc is weakly continuous and, at every time in [ta,tb], hc is a cavity of ho. We distinguish three kinds of dynamic cavities:

- History hc is a *no-exit cavity* of ho if there is no way to escape from hc, except by going through the material of ho itself.
- History hc is a *no-entrance cavity* of ho if there is no way to enter hc, except by going through the material of ho itself.
- History hc is a *persistent cavity* of ho if it is both a no-exit and a no-entrance cavity.

Definition H.C.D.1 defines persistent cavity in terms of no-exit and no-entrance cavities. H.C.A.1 states that no-exit and no-entrance cavities are always spatial cavities and are weakly continuous. H.C.A.2 asserts that if hc is a no-exit cavity of hb and hs is a continuous history that starts inside hc at time ta and is outside hc at a later time tb, then hs overlaps with hb at some intermediate time. H.C.A.3 makes the corresponding assertion for no-entrance cavities. Axiom H.C.A.4 asserts that if histories hc and hb are constant throughout [t1,t2] , and hc is a cavity of hb at t1, then hc is a persistent cavity of hb over [t1,t2]. Axioms H.C.A.5 and 5 assert that the properties NoExitCavity and NoEntranceCavity are inherited by subintervals.

The condition AlwaysIntConn(hs) in H.C.A.2 and H.C.A.3 is needed because these rules do not apply to situations such as illustrated in figure 13, in which history U "seeps through" a point where the cavity in V is in contact with the outside. A physical object cannot do this, of course; however, histories are defined as spatio-temporal entities, and U is a legitimate history.



U seeps through a point in V.

Figure 13: Exception to axiom H.C.A.2 if the condition AlwaysIntConn were omitted

Symbols:
NoExitCavity(t1,t2: *Time*; hc,ho: *History*)
NoEntranceCavity(t1,t2: *Time*; hc,ho: *History*)
PersistentCavity(t1,t2: *Time*; hc,ho: *History*)

Definition:
H.C.D.1 PersistentCavity(t1,t2,hc,hb) ⇔
NoExitCavity(t1,t2,hc,hb) ∧ NoEntranceCavity(t1,t2,hc,hb).

Axioms:
H.C.A.1 [NoExitCavity(t1,t2,hc,ho) ∨ NoEntranceCavity(t1,t2,hc,ho)] ⇒
Lt(t1,t2) ∧ WeaklyContinuous(t1,t2,hc) ∧
$\forall_t$ Leq3(t1,t,t2) ⇒ Cavity(Slice(t,hc),Slice(t,ho)).

H.C.A.2 NoExitCavity(t1,t2,hc,hb) ∧ Continuous(t1,t2,hs) ∧ AlwaysIntConn(t1,t2,hs) ∧
P(Slice(t1,hs),Slice(t1,hc)) ∧ ¬P(Slice(t2,hs),Slice(t2,hc)) ⇒
$\exists_{tm}$ Lt(t1,tm) ∧ Lt(tm,t2) ∧O(Slice(tm,hs),Slice(tm,hb)).

H.C.A.3 NoEntranceCavity(t1,t2,hc,hb) ∧ Continuous(t1,t2,hs) ∧
AlwaysIntConn(t1,t2,hs) ∧
¬P(Slice(t1,hs),Slice(t1,hc)) ∧ P(Slice(t2,hs),Slice(t2,hc)) ⇒
$\exists_{tm}$ Lt(t1,tm) ∧ Lt(tm,t2) ∧O(Slice(tm,hs),Slice(tm,hb)).

H.C.A.4 Constant(t1,t2,hc) ∧ Constant(t1,t2,ho) ∧ Cavity(Slice(t1,hc), Slice(t1,ho)) ⇒
PersistentCavity(t1,t2,hc,ho).

H.C.A.5 NoExitCavity(t1,t2,hc,hb) ∧ Lt(t1,tm) ∧ Lt(tm,t2) ⇒
NoExitCavity(t1,tm,hc,hb) ∧ NoExitCavity(tm,t2,hc,hb)

H.C.A.6 NoEntranceCavity(t1,t2,hc,hb) ∧ Lt(t1,tm) ∧ Lt(tm,t2) ⇒
NoEntranceCavity(t1,tm,hc,hb) ∧ NoEntranceCavity(tm,t2,hc,hb)

7.6.3 Dynamic upright containers

A dynamic upright container is an object that functions as an upright container over a time interval. Specifically the predicate DynamicUprightContainer(ta,tb,ob,hc) asserts that object ob is an upright container with cavity hc (a history) over the interval [ta,tb]. The history hc must be continuous; and at all times tm in [ta,tb], ob must form an upright container with cavity hc which is large enough to contain all the objects that were inside hc at the start time ta (definition H.U.D.1). Under these conditions, the objects inside hc will remain in hc, if there is no external interference (axiom H.U.A.1). Specifically, if ob is a dynamic upright container over the interval with cavity hc, and an object o is inside hc at time ta and outside ob at time tb, then at some time t between ta and tb some object ox was at least partially inside ob and was being manipulated. The object ox may be o itself or may be some other object; e.g. an object being used to scoop up o. In reality there are exceptions to this rule,[12] but it is a good general rule for carrying solid objects in an open container. Axiom H.U.A.1 is a frame axiom in explanation closure form (Schubert, 1990).

Symbols:
DynamicUContainer(ta,tb:*Time*; ob:*Object*; hc:*History*)

Definition:
H.U.D.1 DynamicUContainer(ta,tb,ob,hc) ⇔
Continuous(ta,tb,hc) ∧
$\forall_{tm}$ Leq3(ta,tm,tb) ⇒
UprightContainer(tm,o,Slice(tm,hc)) ∧
Fits(Contents(ta,Slice(ta,hc)),Slice(tm,hc)).

[12] (Davis, 2011) includes an extensive discussion of the exceptions in the case where all the objects involved are rigid.

Axiom:

H.U.A.1 DynamicUContainer(ta,tb,ob,hc) ∧ P(Place(ta,o),Slice(ta,hc)) ∧
¬UContained(tb,o,ob) ⇒
$\exists_{tm,oy}$ Lt(ta,tm) ∧ Lt(tm,tb) ∧ Grasp(tm,oy) ∧
O(Place(tm,oy),Slice(tm,HPlace(hc))).

7.7 Rigid Objects

Rigid objects maintain their shape over time; they are a particularly important and well-behaved kind of object.

7.7.1 Basic Rigid Objects

The place of a rigid object over time is a rigid history (axiom R.O.A.1). We do not define rigid histories geometrically, since that would require a theory of congruence, which we have not developed. For our purposes, the most important property of rigid histories is that any cavity of a time slice of a rigid history is a time slice of a persistent cavity (axiom R.O.A.2)

A history h is rigid and upright if it does not involve any rotations of the vertical axis. If an object is an upright container and its place is a rigid upright history, then it is a dynamic upright container (axiom R.O.A.3).

Symbols:

RigidObject(o: *Object*). — o is a rigid solid object.
RigidHistory(h: *History*) — h is a rigid history.
RigidUprightHistory(ta,tb:*Time;* h: *History*) — h is a rigid history that maintains a vertical axis.

Axioms:

R.O.A.1 RigidObject(o) ⇒ RigidHistory(HPlace(o)).
R.O.A.2 RigidHistory(h) ∧ Leq3(t1,tm,t2) ∧ Cavity(r,Slice(tm,h)) ⇒
$\exists_{hc}$ RigidHistory(hc) ∧ PersistentCavity(t1,t2,hc,h) ∧ r = Slice(tm,hc).
R.O.A.3 RigidUprightHistory(ta,tb,HPlace(o)) ∧ UprightContainer(t,o,rc) ⇒
$\exists_{hc}$ DynamicUContainer(ta,tb,o,hc) ∧ rc=Slice(ta,hc).

7.7.2 Box with lid

The intended meaning of a box with a lid is a pair of rigid objects that form a closed container, where the lid is stably placed on the box, so that, if you move the box, the lid will follow along. We do not axiomatize the conditions necessary for this, which involve both geometric and physical properties of the box and the lid. Rather, we present it as a primitive and use it in some further causal axioms characterizing actions.

Note that the inside of the box-with-lid can be more than the inside of the box by itself viewed as an open container; the lid can arch over the box and enclose more space (figure 14).

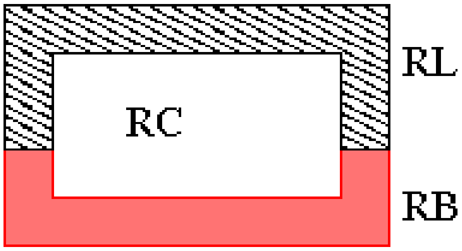


Figure 14: Box with lid

The predicate BoxWithLid(t,ob,ol) asserts that objects ob and ol form a box with lid at time t. BoxWithLidC(t,ol,ob,rc) is the same, with the additional argument rc, the cavity enclosed. BLContained(t,ox,ob) asserts that at time t, object ox is inside a box ob with an unspecified lid.

Axiom R.B.A.1 asserts some basic properties: A box is a pair of rigid objects, not the agent; the lid is stable; and the box and lid form a combined container for a cavity.

Axiom R.B.A.2 states that if a box is motionless and the agent does not grasp the lid, then the lid remains motionless – obviously not always physically true, but true of the way in which a box with lid is standardly used.

Axioms R.B.A.3 is a frame axiom for the BoxWithLid relation; it4 states that a BoxWithLid relation can only be created by manipulating the lid..

Axiom R.B.A.4 states that an agent who is inside a box with a lid cannot grasp the box so as to move it. (However, he may be able to grasp the lid, e.g. to push it off.) It is analogous to axiom M.G.A.3, which asserts the same thing for agents inside closed and open boxes.

Axiom R.B.A.5 asserts that, for any particular pair of objects ob and ol, whether they form a "box with lid" at time t depends entirely on their positions at time t.

Symbols:

BoxWithLid(t: *Time;* ob,ol: *Object*) —

Objects ob,ol physically form a box with lid (thus, ol will move along when ob is moved.)

BoxWithLidC(t: *Time;* ob,ol: *Object;* r: *Region*) —

At time t objects ob,ol form a box with lid with interior rc.

BLContained (t: *Time;* ox,ob: *Object*) —

Object ox is contained in a box ob which has a lid.

Definition:

R.B.D.1 BoxWithLidC(t,ob,ol,rc) ⇔
BoxWithLid(t,ob,ol) ∧

$CombinedContainer(Place(t,ob),Place(t,ol),rc)$..

R.B.D.2 $BLContained(t,ox,ob) \Leftrightarrow$
$\exists_{rc,ol}\ BoxWithLidC(t,ob,ol,rc) \wedge Object(ox) \wedge P(Place(t,ox),rc)$.

Axioms:

R.B.A.1 $BoxWithLid(t,ob,ol) \Rightarrow$
$RigidObject(ob) \wedge RigidObject(ol) \wedge Stable(t,ol) \wedge ob \neq Agent \wedge ol \neq Agent \wedge$
$\exists_{rc}\ CombinedContainer(Place(t,ob),Place(t,ol),rc)$..

R.B.A.2 $BoxWithLid\ (ta,ob,ol) \wedge Released(ta,tb,ol) \wedge Motionless(ta,tb,ob) \Rightarrow$
$Motionless(ta,tb,ol)$.

R.B.A.3 $Lt(ta,tb) \wedge \neg BoxWithLid(ta,ob,ol) \wedge Released(ta,tb,ol) \Rightarrow$
$\neg BoxWithLid(tb,ob,ol)$

.

R.B.A.4 $BLContained(t,Agent,ob) \Rightarrow \neg Grasp(t,ob) \wedge \neg CanGrasp(t,ob)$

R.B.A.5 $BoxWithLid(ta,ob,ol) \wedge$
$Place(tb,ob) = Place(ta,ob) \wedge Place(tb,ol) = Place(ta,ol) \Rightarrow$
$BoxWithLid(tb,ob,ol)$

7.8 Specific Actions

The theory developed so far is too weak to support many of the kinds of inferences we would like to make. In particular, the axioms do not suffice to validate any plans, because there are no axioms giving sufficient conditions for a manipulation to be feasible. In fact, we have not been able to formulate any axioms that give sufficient conditions for manipulation using the kind of general geometric and physical conditions that we have discussed so far in this paper. Rather, we conjecture that, in qualitative reasoning about manipulation, it is necessary to work at a lower level of generality, and develop a collection of more specific theories addressing narrower classes of action.

In this section we sketch the beginning of such a qualitative analysis of manipulations involving containers. We first characterize a class of *safe* manipulations; i.e. manipulations where the effects are predictable and controlled (section 7.8.1). We then formulate a theory for the specific case of safely loading a small object into an upright container (section 7.8.2).

7.8.1 Safe manipulation

To simplify the description of safe manipulations we define a number of predicates. The predicate BoxedIn(t,ox,ob) (definition A.S.D.4) means that ob is a closed container, open container, or box with lid that contains ox. The predicate SafelyMoveWith(t,ox,ob) (definition A.S.D.5) means that ox is an object that reliably moves along with o if o is moved safely. Specifically, either ox is ob itself; or ox is a lid on top of ob; or ox is boxed in ob. The function MovingGroup(t,o) (definition A.S.D.6) is the set of all objects that safely move with o.

The predicate SafeManipulate(ta,tb,o,r) means that object o is manipulated in a safe way during the time interval [ta,tb]. Region r is the region occupied by the moving group of o at time tb. For a manipulation to be safe, the following must hold (axiom A.S.A.1; these are necessary conditions, but not sufficient):

- Object o is the only thing the agent is grasping.
- If o is a closed container containing object ox, then the cavity hc containing o is a no-exit cavity; thus, ox remains inside o (predicate PreserveCContents, definition A.S.D.1).
- If o is an upright container containing object ox, then o is a dynamic upright container; thus, ox remains inside o (predicate PreserveUContents , definition A.S.D.2).
- If o is a box that has a lid, then o is carried vertically upright; (predicate PreserveBoxWithLid , definition A.S.D.3).

If a manipulation of o during [ta,tb] is safe, then it follows from the frame axioms already stated that the objects inside o remain inside o; and any lid on o remains on o.

Axiom A.S.A.2 asserts further that if everything that is not in the moving group of o is stable at time ta, then those objects outside o are all motionless. As discussed in section 7.5.4, this is a special case of a more general principle that the objects outside o evolve independently of the manipulation; but stating that requires a more powerful language.

We say that o is safely movable if the other objects in the world would not interfere with moving it safely; that is, if the agent can get into a position where he can grasp it, then he can move it safely. We do not formally define this predicate; rather, we take SafelyMovable to be a primitive, and we posit (axiom A.S.A.3) that, if two situations are the same except for the position of the agent, then the same objects are safely movable. (A stronger axiom would state that if the objects close to object o are the same at two different times then whether o is safely movable is also the same at the two different times; but that would require a more powerful spatial language than we are using here.)

For example, if o has objects piled on top of it, or is surrounded closely by objects on all sides, then it is not safely movable. We do not here specify geometric conditions sufficient to guarantee that an object is safely movable; rather it is a condition stated in the problem specifications.

Symbols:

PreserveCContents(ta,tb: *Time*; o: *Object*).
PreserveUContents(ta,tb: *Time*; o: *Object*).
PreserveBoxWithLid(ta,tb: *Time*; o: *Object*).
BoxedIn (t: *Time;* ox,ob: *Object*).
SafelyMovesWith(t: *Time;* ox,ob: *Object*)
MovingGroup (t: *Time;* ox: *Object*) → *ObjectSet.*
SafeManipulate(ta,tb: *Time*; o: *Object*; r: *Region*)..
SafelyMovable(t: *Time*; o: *Object*).

Definitions:

A.S.D.1 $\forall_{ta,tb:\ Time;\ o:Object}$ PreserveCContents (ta,tb,o) ⇔
$\forall_{ox,rc}$ ClosedContainer(ta,Singleton(o),rc) ∧ P(Place(ta,ox),rc) ⇒
$\exists_{hc}$ Slice(ta,hc) = rc ∧ NoExitCavity(ta,tb,hc,HPlace(o)).

A.S.D.2

$$\begin{aligned}&\forall_{\text{ta,tb}:\,\mathit{Time};\ \text{o}:\mathit{Object}}\ \text{PreserveUContents (ta,tb,o)} \Leftrightarrow\\ &\quad \forall_{\text{ox,rc}}\ \text{UprightContainer(ta,o,rc)} \land \text{P(Place(ta,ox),rc)} \Rightarrow\\ &\qquad \exists_{\text{hc}}\ \text{Slice(ta,hc)} = \text{rc} \land \text{DynamicUContainer(ta,tb,o,hc)}.\end{aligned}$$

A.S.D.3

$$\begin{aligned}&\forall_{\text{ta,tb}:\,\mathit{Time};\ \text{o}:\mathit{Object}}\ \text{PreserveBoxWithLid (ta,tb,o)} \Leftrightarrow\\ &\quad [\exists_{\text{ol}}\ \text{BoxWithLid (ta,ob,ol)}] \Rightarrow \text{RigidUprightHistory(ta,tb,HPlace(ob))}.\end{aligned}$$

A.S.D.4

$$\begin{aligned}&\text{BoxedIn(t,ox,ob)} \Leftrightarrow\\ &\quad \text{CContained(t,ox,Singleton(ob))} \lor \text{UContained(t,ox,ob)} \lor\\ &\quad \text{BLContained(t,ox,ob)}\end{aligned}$$

A.S.D.5

$$\begin{aligned}&\forall_{\text{t}:\,\mathit{Time};\ \text{ox,ob}:\mathit{Object}}\ \text{SafelyMovesWith(t,ox,ob)} \Leftrightarrow\\ &\quad \text{ox} = \text{ob} \lor \text{BoxedIn(t,ox,ob)} \lor \text{BoxWithLid(t,ob,ox)}.\end{aligned}$$

A.S.D.6

$$\begin{aligned}&\forall_{\text{ta};\mathit{Time};\text{o,ox}:\ \mathit{Object}}\ \text{Element(ox,MovingGroup(t,o))} \Leftrightarrow\\ &\quad \text{SafelyMovesWith(t,ox,o)}.\end{aligned}$$

Axioms:

A.S.A.1

$$\begin{aligned}&\text{SafeManipulate(ta,tb o,r)} \Rightarrow\\ &\quad \text{Grasps(ta,tb,o)} \land\\ &\quad [\forall_{\text{ox}:\mathit{Object};\ \text{tm}:\mathit{Time}}\ \text{ox} \neq \text{o} \land \text{Leq3(ta,tm,tb)} \Rightarrow \neg\text{Grasp(tm,ox)}] \land\\ &\quad \text{OSPlace(tb,MovingGroup(ta,o),r)} \land\\ &\quad \text{PreserveCContents (ta,tb,o)} \land \text{PreserveUContents (ta,tb,o)} \land\\ &\quad \text{PreserveBoxWithLid (ta,tb,o)}\end{aligned}$$

A.S.A.2

$$\begin{aligned}&\text{SafeManipulate(ta,tb o,r)} \land\\ &\quad [\forall_{\text{ox}:\mathit{Object}}\ \text{ox} = \text{Agent} \lor \text{Element(ox,MovingGroup(ta,o))} \lor\\ &\qquad \text{Stable(ta,ox)}] \Rightarrow\\ &\quad [\forall_{\text{ox}:\ \mathit{Object}}\ \text{ox} = \text{Agent} \lor \text{Element(ox,MovingGroup(ta,o))} \lor\\ &\qquad [\text{Motionless(ta,tb,ox)} \land \text{StableThroughout(ta,tb,ox)}]].\end{aligned}$$

A.S.A.3

$$\begin{aligned}&\text{SafelyMovable(ta,o)} \land\\ &\quad [\forall_{\text{ox}}\ \text{ox} \neq \text{Agent} \Rightarrow \text{Place(tb,ox)} = \text{Place(ta,ox)}] \Rightarrow\\ &\quad \text{SafelyMovable(tb,o)}\end{aligned}$$

7.8.2 Loading an Upright Container

We now give a theory for the specific action of loading a small object into an upright container. We define two actions. The simple action PutInUC(ox,ob) is the action of safely putting an object ox into an open container oc (definition A.L.D.1). The compound action LoadUprightContainer(ox,ob) is a sequence of three steps: The agent travels to a position where he can grasp ox, loads ox into the open container ob, and then moves his hand out of ob.

We posit two feasibility axioms associated with these. Axioms A.L.A.1 asserts that it is feasible to load ox into ob if ox can be grasped and safely moved, and the agent can reach the inside of ob, and the current contents of ob together with ox are small as compared to the inside of ob, and everything is stable (so that we can be sure that nothing will fall to block the path). Axiom A.L.A.2 states that, if the agent can initially travel to some destination rx which is fully outside the container ob and then loads ox into ob, then the agent can still travel to rx. It is fairly easy to find exceptions to A.L.A.1, and it is possible, though not easy, to find exceptions to A.L.A.2, but they are both quite good rules of thumb.

Symbol:

$\text{Reachable}(\text{t}: \textit{Time};\ \text{r}: \textit{Region})$
$\text{PutInUC}(\text{ox},\text{ob}: \textit{Object}) \rightarrow \textit{Action}.$
$\text{LoadUprightContainer}(\text{ox},\text{ob}: \textit{Object}) \rightarrow \textit{Action}$

Definitions:

A.L.D.1 $\forall_{\text{ta,tb}: \textit{Time};\ \text{ox,ob}: \textit{Object}}\ \text{Occurs}(\text{ta},\text{tb},\text{PutInUC}(\text{ox},\text{ob})) \Leftrightarrow$
$\exists_{\text{rc,rx}}\ \text{UprightContainer}(\text{ta},\text{ob},\text{rc}) \wedge \text{P}(\text{rx},\text{rc}) \wedge$
$\text{SafeManipulate}(\text{ta},\text{tb},\text{ox},\text{rx}) \wedge$
$\text{PartiallyContained}(\text{Place}(\text{tb},\text{Agent}),\text{Place}(\text{tb},\text{ob}))$

A.L.D.2 $\forall_{\text{ta,tb}: \textit{Time};\ \text{ox,ob}: \textit{Object}}\ \text{Occurs}(\text{ta},\text{tb},\text{LoadUprightContainer}(\text{ox},\text{ob})) \Leftrightarrow$
$\exists_{\text{r1,r3}}\ \text{FullyOutside}(\text{r3},\text{Place}(\text{ta},\text{ob})) \wedge$
$\text{Occurs}(\text{ta},\text{tb},\text{Sequence}(\text{TravelTo}(\text{r1}),$
$\text{Sequence}(\text{PutInUC}(\text{ox},\text{ob}),\ \text{TravelTo}(\text{r3})))).$

Loading object ox into open upright container ob is the sequence of travelling to a place where ox can be grasped, moving ox inside ob and then withdrawing the manipulator out of ob. The container ob remains motionless throughout.

A.L.D.3 $\text{Reachable}(\text{ta},\text{r}) \Leftrightarrow$
$\exists_{\text{rx}:\textit{Region}}\ \text{IntConn}(\text{RUnion}(\text{rx},\text{r})) \wedge \text{Feasible}(\text{t},\text{TravelTo}(\text{rx})).$

Region r is reachable at time t if it is feasible for the agent to travel to a position rx such that r ∪ rx is interior connected.

Axiom:

A.L.A.1 $\text{UprightContainer}(\text{ta},\text{ob},\text{rc}) \wedge \text{CanGrasp}(\text{ta},\text{ox}) \wedge \text{Reachable}(\text{ta},\text{rc}) \wedge$
$\text{SafelyMovable}(\text{ta},\text{ox}) \wedge \text{AllStable}(\text{ta}) \wedge \text{SmallObject}(\text{ob}) \wedge$
$\text{SmallSet}(\text{Union}(\text{UContents}(\text{ta},\text{ob}),\text{MovableGroup}(\text{ta},\text{ox})),\ \text{rc}) \Rightarrow$
$\text{Feasible}(\text{ta},\text{PutInUC}(\text{ox},\text{ob})).$

Feasibility axiom: If ob is an upright container with cavity rc, the agent can grasp ox, ox together with the current contents of rc is small as compared to rc, ox is safely movable, ob is stable, and the agent can reach inside rc, then the agent can load ox into ob.

A.L.A.2 $\text{Occurs}(\text{ta},\text{tb},\text{PutInUC}\ (\text{o},\text{ob})) \wedge \text{Feasible}(\text{ta},\text{TravelTo}(\text{rx})) \wedge$
$\text{FullyOutside}(\text{rx},\text{Place}(\text{ta},\text{ob})) \Rightarrow$
$\text{Feasible}(\text{tb},\text{TravelTo}(\text{rx})).$

8. Inferences

We come at last to our example inferences.

Note that, if an inference can be made, then any logically equivalent inference can be made with a slight adaptation to the proof. For instance, Scenario 1 states that if box Ob1 is a rigid object and, at time Ta1 contains object Ox1 as a closed container, then Ox1 is still in Ob1 at any later time; this is a prediction problem. Equivalently, one can infer that, if at time Ta1, Ob1 is a closed container containing Ox1, and at time Tb1, Ox1 is not inside Ob1, then Ob1 is not a rigid object; this is a problem of inferring object characteristics from observations made over time.

8.1 Inference 1

Qualitative prediction. If Ob1 is a rigid object and it is a closed container containing object Ox1, then Ox1 remains inside Ob1 (figure 16).

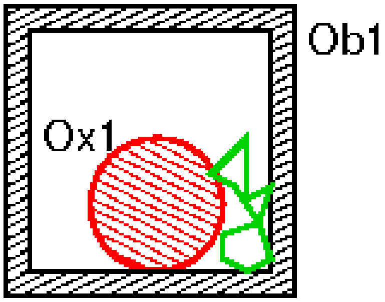


Figure 16: Inference 1.
Here and in some of the other diagrams illustrating inferences, we show some additional objects in green, to emphasize that the presence or absence of these does not affect the inference.

Symbols:
Ox1 → *Object* — Some stuff.
Ob1 → *Object* — A box.
Ta1,Tb1 → *Time* — Times.

Given:
C.1.A.1 RigidObject(Ob1).
C.1.A.2 CContained(Ta1,Ox1,Singleton(Ob1)).
C.1.A.3 Lt(Ta1,Tb1).

Infer: CContained(Tb1,Ox1,Singleton(Ob1)).

Sketch of Proof: Object Ob1 lies inside a cavity of Ox1 (O.C.D.1, O.R.D.1). Since Ox1 is a rigid object, there is a corresponding persistent cavity (R.O.A.1, R.O.A.2). If Ox1 were go to from inside the cavity to outside the cavity, it would have to spatially overlap Ob1 (H.C.A.2), but this is impossible because they are different objects (O.T.A.5). (Place(T1a,Ob1) contains Place(Ta1,Ox1), and it is a geometric theorem that a region cannot contain itself.)

8.2 Inference 2

Qualitative prediction. If Ob2b is a rigid object and a closed container containing Ob2a, and Ob2a is a closed container (not necessarily rigid) containing object Os2, then Os2 will remain inside Ob2b (figure 17).

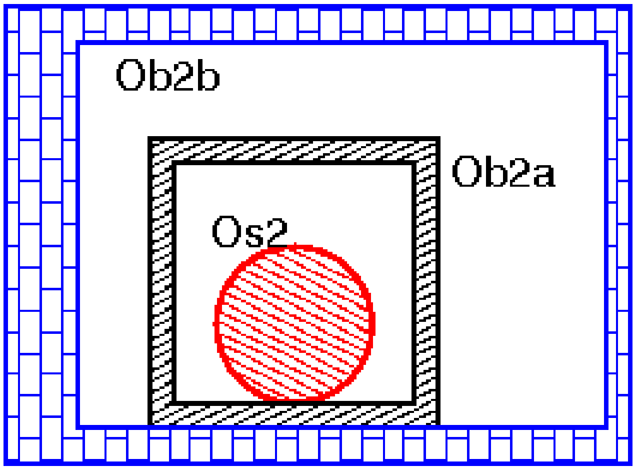


Figure 17: Inference 2

Symbols:
Os2 → *Object* — Some stuff.
Ob2a → *Object* — Inner container.
Ob2b → *Object* — Outer box.
Ta2,Tb2 →*Time* —Times.

Given:
C.2.A.1 RigidObject(Ob2b).
C.2.A.2 CContained(Ta2,Os2,Singleton(Ob2a)).
C.2.A.3 CContained(Ta2,Ob2a,Singleton(Ob2b)).
C.2.A.4 Ordered(Ta2,Tb2).
Infer: CContained(Tb2,Os2a,Singleton(Ob2a)).

Sketch of Proof: By axiom S.C.A.1, spatial closed containment is transitive; hence Os2a is contained in Ob2b. The result then follows from scenario 1.

As an illustration of how carefully these inferences have to be formulated, figure 18 illustrates that this inference is not valid if the inner container is an open container. Let Oa be the red, U-shaped region with hatching;; let Ob be the blue U-shaped region with an internal cavity containing Oa and let Os be the green ball. Then Oa contains Os as an open container and Ob contains Oa as a closed container, but Ob does not contain Os as a closed container.

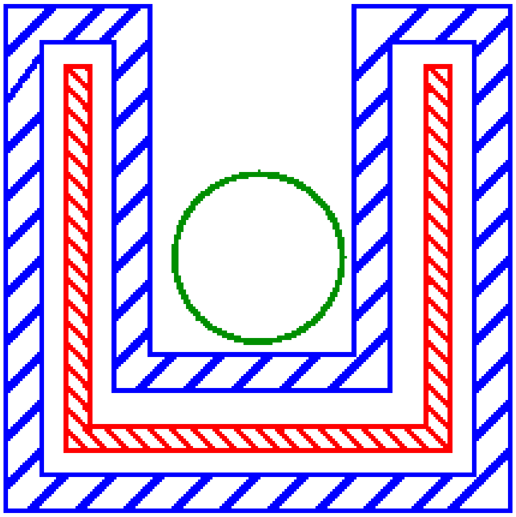


Figure 18: Alternative to scenario 2

Note also that the condition that the two time points Ta2 and Tb2 are ordered is necessary; otherwise, they could be on two completely unrelated time lines.

8.3 Inference 3

If the situation depicted in figure 3 above is modified so that the red region is flush against the barriers , then the ball must reach the red region before it can reach the green region.

Symbols:
Os3 → *Object*. Movable object.
Ob3 →*Object*. Fixed object.
RRed, RGreen, RInside → *Region.* Two target regions.
Ta3, Tb3 → *Time.*

Given:
C.3.A.1 Fixed(Ob3).
C.3.A.2 CombinedContainer(Place(Ta,Ob3), RRed, RInside)
C.3.A.3 P(Place(Ta3,Os3),RInside).
C.3.A.4 Outside(RGreen, RUnion(Place(Ta3,Ob3),RRed)).
C.3.A.5 P(Place(Tb3,Os3),RGreen).
C.3.A.6 Lt(Ta3,Tb3).
C.3.A.7 Os3 ≠ Ob3.

Infer: $\exists_{tm}$ Lt(Ta3,tm) ∧ Lt(tm,Tb3) ∧ O(Place(tm,Os3),RRed).

Sketch of Proof: Let Ru be the spatial union of the red region plus the object Ob3. This is a closed container containing Ob3 at the start. By the same argument as in scenario 1, if object Os3 goes from inside this container to outside it in the green region, it must overlap the union at some time in between. Since it cannot overlap Ob3, it must overlap the red region (axiom S.B.A.4).

There are two differences between the problem as analyzed here, on the one hand, and the problem as presented by Smith et al. (2013) to their subjects, on the other. First, the geometry has been altered. In Smith et al.'s experiment, shown in figure 3 above, the red region is slightly separated from the barriers. Second, and more importantly, the physics is different; the experiment deals with a ball bouncing autonomously, whereas our physics objects move only when manipulated or when falling.

These differences become important if one considers alternative versions of the problem in which the red region is further and further from the barriers. If the red region is quite close to the barriers, one can reason that there is no room for the ball to "squeeze through" the gap between the red region and the barriers. This could be added to our theory with only a slight extension of the geometric language, plus the specification that the ball is a rigid object. If the red region is further away, then one has to reason that the ball must be moving rightward when it exits the region contained by the barriers, and therefore can never reach the green region. (In figure 3 as drawn, the diameter of the ball is almost exactly equal to the width of the gap, so it is not clear which of these applies.) Carrying out the inference about direction of motion in the second case would require a very substantial extension to our spatial and physical theory, since our spatial language has no representation of direction other than the vertical, and our physical theory incorporates no idea of momentum. Adding a theory of direction to the spatial theory is straightforward and not problematic. Adding a useful qualitative theory of momentum sufficient to this inference to the physical theory in a principled way would be a substantial project, though formulating *ad hoc* rules sufficient for this particular inference would be easy enough. (Forbus' (1979) FROB program developed a qualitative theory of momentum, but only for a point object moving among fixed obstacles in two-dimensional space divided into zones along the cardinal directions.)

8.4 Inference 4

If Ox4 is outside upright container Ob4, and the current contents of Ob4 together with Ox4 are much smaller than the interior of Ob4, and the agent can reach and move Ox4 and can reach into Ob4, then (a) the agent can load Ox4 into Ob4; (b) if the agent does load Ox4 into Ob4, then Ox4 and all its initial contents and all the initial contents of Ob4 will end up in a stable state inside Ob4 (figure 19).

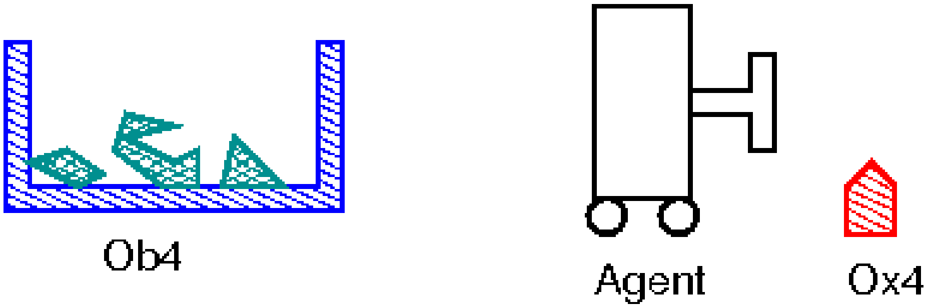


Figure 19 :Inference 4: Starting state

Symbols:
Ob4 → *Object.* Object being loaded.
Ox4 → *Object.* Open container.
Rc4 → *Region.* Inside of Ox4.

Ta4 → *Time.*

Given:

C.4.A.1 UprightContainer(Ta4,Ob4,Rc4).
C.4.A.2 FullyOutside(Place(Ta4,Ox4),Place(Ta4,Ob4)).
C.4.A.3 SmallSet(Union(UContents(Ta4,Ob4),MovingGroup (Ta4,Ox4)),Rc4).
C.4.A.4 AllStable(Ta4).
C.4.A.5 EmptyHanded(Ta4).
C.4.A.6 Graspable(Ta4,Ox4).
C.4.A.7 Reachable(Ta4,Rc4).
C.4.A.8 Ox4 ≠ Agent ≠ Ob4.
C.4.A.9 FullyOutside(Place(Ta4,Agent),Place(Ta4,Ob4)).
C.4.A.10 SafelyMovable(Ta4,Ox4).
C.4.A.11 SmallObject(Ob4).
C.4.A.12 ¬BoxedIn(Ta4,Agent,Ob4).
C.4.A.13 NoPartialContents(Ta4,Ob4).

Infer: 4.a: Feasible(Ta4, LoadUprightContainer(Ox4,Ob4)).

Sketch of Proof: Axioms A.L.D.1, A.L.D.2, and T.A.D.2 are used to expand the meaning of LoadUprightContainer: The agent must first travel to a position where he can grasp Ox4, then manipulate it so that it is inside Ob4, then withdraw his hand from Ob4. It follows from M.F.D.4 that the initial travel is feasible. It follows from A.L.A.1 that, after travelling to grasp Ox4, it will be feasible for him to put Ox4 inside Ob4. It follows from A.L.A.2 that, after putting Ox4 inside Ob4, it will be feasible for him to withdraw his hand. The details of the proof are quite long, because it takes work to establish that the conditions of A.L.A.1 and .2 will be met at the times in question. In particular, it is necessary to carry out many different frame inferences, stating that important conditions do not change while these actions are being executed.

Infer: 4.b: $\forall_{tb}$ Occurs(Ta4,tb ,LoadUprightContainer(Ox4,Ob4)) ⇒
$\exists_{tc}$ Occurs(tb ,tc,StandStill) ∧ AllStable(tc) ∧
UContents(tc,Ob4) =
Union(UContents(Ta4,Ob4),MovingGroup(Ta4,Ox4)),

Sketch of Proof: From A.L.D.1, M.S.A.1, A.S.A.1, A.S.D.5, A.S.D.6 it follows that, after the action PutInUC(Ox4,Ob4) (the second step of LoadUprightContainer(Ox4,Ob4)), the object Ox4 will be inside the upright container Ob4. Axiom M.S.A.1 asserts that, after the agent has released Ox4, the world will eventually attain a stable state. Axiom M.R.A.4 asserts that, while waiting for the world to attain stable state, all the objects in the upright box Ob4 will remain inside the upright box. As with part 4.a, projecting forward all the states of the system requires many steps.

8.5 Inference 5

Let Ob5 and Ol5 be a box with lid at time Ta5, and let Os5 be an object inside the box. Assume that the agent is outside the box at time Ta5. If Os5 is somewhere else at time Tb5, and the box is fixed throughout [Ta5,Tb5] then the lid must move at some time in between Ta5 and Tb5 (figure 20).

Ob5 → *Object*. Box
Ol5 → *Object*. Lid.
Os5 → *Object*. Stuff.
Rc5 → *Region*. Inside of box with lid.
Ta5, Tb5 → *Time*.

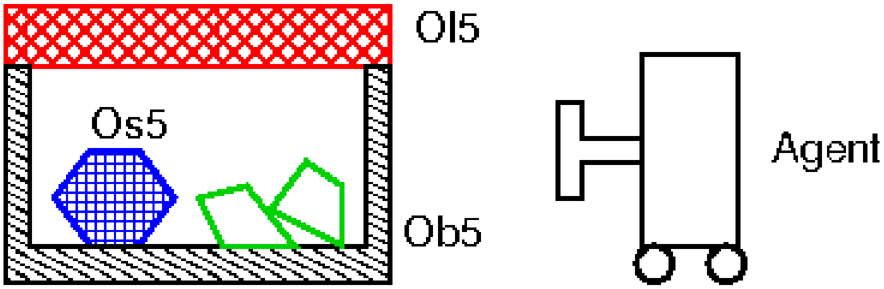


Figure 20: Inference 5

Given:
C.5.A.1 BoxWithLidC(Ta5,Ob5,Ol5,Rc5).
C.5.A.2 P(Place(Ta5,Os5),Rc5).
C.5.A.3 ¬P(Place(Ta5,Agent),Rc5).
C.5.A.4 AllStable(Ta5).
C.5.A.5 Place(Ta5,Os5) ≠ Place(Tb5,Os5)
C.5.A.6 Constant(Ta5,Tb5,HPlace(Ob5))

Infer: ¬Motionless(Ta5,Tb5,Ol5).

Sketch of proof: Proof by contradiction. Suppose that the lid remains motionless. Then the box and the lid form a closed container (definition R.B.D.1), so, by an argument analogous to scenario 1, neither the agent nor any object that the agent can reach can get inside the box. Therefore, the set of objects in the box is static causally isolated (definitions M.R.D.1, M.R.D.2), and therefore all the objects are motionless (axiom M.R.A.2), contradicting C.5.A.5.

8.6 Formal proofs and automated verification

Complete human-written formal proofs of Inference 1-5 in a natural-deduction format may be found at http://www.cs.nyu.edu/faculty/davise/containers/Proofs.html. These proofs have been verified using the SPASS theorem prover (Weidenbach, et al., 2009); the inputs and outputs of the verification are linked at the same web site.

It may be noted that a substantial amount of human labor (some number of man-months) was involved in carrying out the automated verification, even beyond the manual construction of the formal proofs. When we had completed the construction of the first draft of the formal proofs, we thought, optimistically, that the process of verification would be easy and straightforward; we would be able to divide the proof into large chunks, code up each chunk, and run it through SPASS. In practice, the gaps in the proofs we had written were so frequent and so large that we often ended up generating separate SPASS runs for each step in the formal proofs. The problems that we encountered fell into three categories:

- Errors in translating the human-written proofs into valid SPASS input. In particular, the human proofs deliberately omit sortal preconditions and often omit necessary temporal axoms. These had to be restored in preparing the SPASS input.
- Gaps in the human-written proofs. This was the most frequent form of problem. The human-written proofs often either omitted some of the justifications needed for a step in the proof or omitted some of the steps needed to complete a proof. Several times, additional lemmas requiring proofs of a dozen or more steps needed to be added.
- Gaps and errors in the actual theory. The most important outcome of this work on automated verification was to find a half-dozen mistakes in the axioms as they were then written in the draft of section 7. Most of these were fairly trivial; the wrong variable was used, arguments were in the wrong order, a simple object variable o was used in a context where Singleton(o) was required. One error was more substantive; completing the proof required adding an additional axiom.

It would certainly be possible to write tools, such as a sort-checking interface, that would very much reduce the labor involved in completing the automated proofs, or to use existing tools, and if this kind of work is done on a larger scale, that would certainly be worthwhile.

9. Consistency

One might well wonder, seeing the complexity of the theory developed here, whether there is any hope of establishing that it is consistent. As we shall see, it is extremely easy to show that it is consistent, in a trivial sense, and not difficult to show that it is consistent in a non-trivial sense, but neither of these answers the question that one presumably has in mind. However, it is not at all easy to say exactly what the “right” question is, let alone to answer it.

First, there is an entirely trivial sense in which the theory is consistent. One establishes that a theory is consistent by presenting a model for it. It is easily checked that the following model satisfies all the axioms in section 7 of the paper.

Model 1: There is a single *Object*, namely the agent. (The agent has to exist, because he is named by a constant symbol.) There is a single *History,* namely HPlace(Agent). There are two *ObjectSets*: Null and Singleton(Agent). The set of *Actions* is StandStill, PutInUC(Agent,Agent), LoadUprightContainer(Agent,Agent), and all finite sequences of these three. These are the only entities that exist; the sorts *Time* and *Region* both denote the empty set.[13] If you go back and check, you will see that there are no axioms that require that any times exist; and if no times exist, there are no axioms that require that any regions exist.

[13] We are assuming here a sort-sensitive semantics, in which $F(x_1 \dots x_n)$ exists only if $x_1 \dots x_n$ satisfy the sort declaration of F. Otherwise, the model will have to specify values for bogus terms like HPlace(StandStill). The values of these can consistently be assigned arbitrarily; e.g. these can all denote Null.

Of course, this is a ridiculous model, but it is worthwhile being specific about why it is ridiculous. There are two obvious problems. First, it doesn't at all match the description of the model given in the text; we had said that *Times* exist within a branching continuous time line; that *Regions* are topologically regular regions of $\mathbb{R}^3$, and so on. Second, the model is completely inconsistent with assumptions of inferences 1-5 of section 8.

In view of these objections, we can reformulate the objective as follows: Can we construct a model that incorporates branching continuous time and three-dimensional geometry, and that satisfies both the axioms of section 7 and the boundary conditions of inferences 1-5? The answer is again yes, and is still not very difficult, though no longer trivial. We will sketch below a model (more precisely, a class of models) satisfying scenario 1; the construction of the models satisfying the other scenarios, individually or all together, is analogous, though somewhat more involved.

Model 2: Intuitively, this model will consist of the two objects Ob and Ox and the Agent remaining motionless for all time.

Formally: The denotation of sort *Time* is $\mathbb{R}$*,* the real line. The denotation of sort *Region* is some set of topologically regular, bounded subsets of $\mathbb{R}^3$. There are four specific regions Rb, Rc, Rx, and RAgent. Rb is a closed container; Rc is a cavity of Rb; Rx is a connected region which is a subset of Rc, RAgent is a connected region which is disjoint from Rb and Rc. Rb, Rc, Rx, and RAgent are elements of *Region*, and the class *Region* is closed under finite union; otherwise, the choice of which topologically regular, bounded subsets of $\mathbb{R}^3$ to include in *Region* is arbitrary. The sort *History* is some class of functions from *Time* to *Region.* For every region r, the sort *History* includes the constant function from *Time* to r; which other functions are included in *History* is arbitrary.

The sort *Object* is equal to the set {Ob, Ox, Agent }. The sort *ObjectSet* is the set of all subsets of *Object.* The primitive actions are StandStill, PutInUC(o1,o2), and LoadUprightContainer(o1,o2), where o1 and o2 are two of the objects (not necessarily distinct). The sort Action includes all finite sequences of primitive actions.

The purely temporal primitives, the purely spatial primitives, the spatio-temporal primitives associated with histories, and the set-theoretic primitives have their standard interpretations. Predicate Occurs(ta,tb,a) holds if and only if ta < tb and a is a sequence of StandStill. For all times t, Place(t,Agent)=RAgent, Place(t,Ob)=Rb, Place(t,Ox)=Rx. The extension of predicate OSPlace(t,s,r) is then determined by definition O.T.D.1. FeasiblePlace(o,r) holds if and only if [o=Agent and r =RAgent] or [o=Ox and r=Rx] or [o=Ob and R=Rb]. The extension of predicates SmallObject and SmallSet is empty. Predicate Isolates(t,sp,sx) holds for all t, for sp={ Ob } and sx=Ox and for no other values. Grasps(t,o) and Fixed(o) do not hold for any values. Stable(t,o) holds for all times t and all objects o. Trajectory(ra,rb,rw) holds if ra=rb and rw is a superset of ra. DynamicUContainer(ta,tb,ob,hc) and BoxWithLid(t,ob,ol) hold for no values. RigidObject(o) holds for object ob and ox. The extensions of the remaining predicates are determined by the definitions and axioms. **End of description of model 2.**

The models satisfying Scenarios 2, 3, and 5 are analogous. In the models for scenario 2, the objects are the agent and the named objects and everything remains still forever. In the model for scenario 3, the objects are the agent, the ball, and the frame. The agent always stands still. The frame is fixed. The ball moves on a feasible path for a while and then settles down. In the model for scenario 5, all the objects remain motionless.

The model for scenario 4 is more complicated. There are four objects: the agent, the box Ob4, the object Ox4, and the ground, which is fixed. The model specifies some geometry for these that would allow the agent to put the object into the box. There is one time line in which the agent goes over to Ox4, grasps it, carries it over to Ob4, puts it inside Ob4, and releases it; and then Ox4 clatters to the bottom of Ob4. From each time point on this line, there is an alternative branch of time structure in which the agent stands still forever. From each time point t on the original line or on these branches, if the agent is grasping Ox4 at t then there is an alternative branch of the time structure in which the agent releases Ox4, Ox4 falls to the ground, and then eventually attains a stable state.

Full formal specifications of these models are available from the authors on request.

The construction of these kinds of models certainly bring us closer to establishing that the theory is OK, in the sense we are looking for, but there is still a long way to go. All we have done is to establish that these specific scenarios are consistent with our theory. Of course, we could easily generalize these somewhat and add more scenarios, but still all we would have is a collection of special cases. What we are really looking for is a general theorem that any “reasonable” problem specification is consistent with the axioms, and does not entail overly strong; but it is not at all clear what constitutes a reasonable problem specification. Also, and even more vaguely, we would like some assurance that our theory is “forward-compatible” in the sense if we fill in reasonable theories of stability, of feasible manipulations, and of the other parts of our theory that we have left entirely unspecified, the result will still be consistent. But again it is hard to see how one would formally state such a meta-theorem, let alone prove it.

10. Related Work

There is much previous AI work on physical reasoning with partial information, especially under the rubric of "qualitative reasoning" (QR) in the narrow sense (Bobrow, 1985). This work has primarily focussed on qualitative differential equations (Kuipers, 1986) or similar formalisms (Forbus, Qualitative process theory, 1985) (de Kleer & Brown, 1985). The current project is broadly speaking in the same spirit; however, it shares very little technical content, because of a number of differences in domain. First, our theory uses a much richer language of qualitative spatial relations than in most of the QR literature. Systems considered in the QR literature tend to involve either no geometry (e.g. electronic circuits (de Kleer and Brown, 1985); highly restricted geometry (e.g. the geometry of linkages (Kim, 1992)); or fully specified geometries (e.g. (Faltings, 1987)). Second, the problems considered in the QR literature involved primarily the internal evolution of physical systems; exogenous action was a secondary consideration. In our domain, almost all change is initiated by the action of an agent. Finally, the QR literature is primarily, though not exclusively, focused on prediction; our theory is designed with the intent of supporting inference in many different directions.

More directly relevant to our project is the substantial literature on qualitative spatial reasoning, initiated by the papers of Randell, Cui, and Cohn (1992) and of Egenhofer and Franzosa (1991) and extensively developed since (Cohn & Renz, 2008). In particular, we use the RCC-8 language of topological relations between regions as the basis of our spatial representation; the concept of a closed container can be defined in that language. However, the theories of open containers and of open upright containers, and the theory of temporally evolving containers are largely new here.

In previous work (Davis, 2008) (Davis, 2011) we developed representation languages and systems of rules to characterize reasoning about loading solid objects into boxes and carrying objects in boxes, and pouring liquids between open containers. The theory discussed in this paper differs in two major respects. First, the earlier work developed moderately detailed dynamic theories of rigid solid object and of liquids. In this paper, the dynamic theories apply across a much wider range of materials, and therefore are much less detailed. Second, the previous work made arbitrary use of set theory, geometry, and real analysis in constructing the proofs; that is, considerations of both effective implementation and cognitive plausibility were entirely ignored in favor of representational and inferential adequacy. The current project aims at a theory that is both effectively implementable and cognitively realistic, sacrificing expressive and inferential power where necessary.

Kim (1993) developed a system that carried out qualitative predictions of the motions of liquids in response to the motions of pistons. She also included in her model a special case of solids being acted on by liquids, namely the opening and closing of one-way valves. Her theory was mostly concerned with qualitative reasoning about pressures and forces between solids and liquids, and thus quite disjoint from the issues considered here. She did not discuss moving the container as a whole, containers with any non-rigidity other than pistons and valves, or any containment relations other than that a solid container and liquid contents. Both the geometric and physical language of this system were quite limited.

A study by Smith, Dechter, Tenenbaum, and Vul (2013) studies the way in which experimental participants reason about a ball bouncing among obstacles. They demonstrate that, though in many circumstances, subjects' responses are consistent with a theory that they are simulating the motion of the ball, under some circumstances where qualitative reasoning easily supplies the answer, they can answer much more quickly than the simulation theory would predict. For example, when presented with the situation shown in figure 3 and asked, "Which region does the ball reach first: the red region or the green region?" they can quickly answer "the red region". As it happens, all of the instances of qualitative reasoning they discuss can be viewed as some form of reasoning about containment. As discussed in section 8.3, our knowledge base supports a modified form of this particular inference (assuming that the ball is itself the agent or is being moved by the agent.)

11. Conclusions and future work

Human commonsense physical reasoning is strikingly flexible in its ability to deal with radically incomplete problem specifications and incomplete theories of the physics of the situation at hand. We have argued that an appropriate method for achieving this flexibility in an automated system would be to use a knowledge-based system incorporating rules spanning a wide range of specificity. As an initial step, we have formulated some of the axioms that would be useful in reasoning about manipulating containers, and we have shown that these axioms suffice to justify some simple commonsense inferences.

In future work on this project, we plan to expand the knowledge base to cover many more forms of qualitative reasoning about containers, and to expand the collection of commonsense inferences under consideration. We will also attempt to implement the knowledge base within an automated reasoning system that can carry out the inferences from problem specification to conclusion.

Acknowledgements

Thanks to Angelica Chan and Casey McGinley for their participation in the project.

References

Battaglia, P., Hamrick, J., & Tenenbaum, J. (2013). Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences, 110*(45), 18327-18332.

Bobrow, D. (1985). *Qualitative Reasoning about Physical Systems.* Cambridge, Mass.: MIT Press.

Casati, R., & Varzi, A. (1994). *Holes and Other Superficialities.* Cambridge, Mass:: MIT Press.

Cohn, A., & Renz, J. (2008). Qualitative Spatial Representation and Reasoning. In F. van Harmelen, V. Lifschitz, & B. Porter, *Handbook of Knowledge Representation* (pp. 551-596). Amsterdam: Elsevier.

Davis, E. (1998). The naive physics perplex. *AI Magazine, 19*(4), 51-79.

Davis, E. (2001). Continuous shape transformation and metrics on regions. *Fundamenta Informaticae, 46*(1-2), 31-54.

Davis, E. (2008). Pouring liquids: A study in commonsense physical reasoning. *Artificial Intelligence, 172,* 1540-1578.

Davis, E. (2011). How does a box work? *Artificial Intelligence, 175*(1), 299-345. doi:http://dx.doi.org/10.1016/j.artint.2010.04.006

Davis, E., & Marcus, G. (2014, December). *The Scope and Limits of Simulation in Cognition.* Retrieved from arXiv: http://arxiv.org/abs/1506.04956

Davis, E., & Marcus, G. (2015). Commonsense Reasoning and Commonsense Knowledge in Artificial Intelligence. *Communications of the ACM, 58*(9), 92-103.

Davis, E., & Marcus, G. (2016, April). The Scope and Limits of Simulation in Automated Reasoning. *Artificial Intelligence*, *233*, 60-72.

de Kleer, J., & Brown, J. (1985). A qualitative physics based on confluences. In D. Bobrow, *Qualitative Reasoning about Physical Systems* (pp. 7-84). Cambridge: MIT Press. doi:http://dx.doi.org/10.1016/0004-3702(84)90037-7

Egenhofer, M., & Franzosa, R. (1991). Point-set topological spatial relations. *Intl. Journal of Geographical Information Systems, 5*(2), 161-174.

Faltings, B. (1987). Qualitative Kinematics in Mechanisms. *IJCAI,* (pp. 436-442).

Forbus, K. (1979). *A study of qualitative and geometric reasoning in reasoning about motion.* Tech. Report 615, MIT AI Lab.

Forbus, K. (1985). Qualitative process theory. In D. Bobrow, *Qualitative Reasoning about Physical Systems* (pp. 85-168). Cambridge, Mass.: MIT Press. doi:http://dx.doi.org/10.1016/0004-3702(84)90038-9

Hayes, P. (1977). In defense of logic. *IJCAI,* (pp. 559-565).

Hayes, P. (1979). The naive physics manifesto. In D. Michie, *Expert Systems in the Microelectronic Age.* Edinburgh: Edinburgh University Press. From http://aitopics.org/publication/naive-physics-manifesto

Hayes, P. (1985). Ontology for Liquids. In J. Hobbs, & R. Moore, *Formal Theories of the Commonsense World.* Ablex.

Hegarty, M. (2004). Mechanical reasoning by mental simulation. *Trends in Cognitive Science, 8*(6), 280-285. doi:http://dx.doi.org/10.1016/j.tics.2004.04.001

Hespos, S., & Baillargeon, R. (2001). Reasoning about containment events in very young infants. *Cognition, 78*, 207-245.

Kim, H. (1992). Qualitative Kinematics of Linkages. In B. Faltings, & P. Struss, *Recent Advances in Qualitative Physics.* Cambridge, Mass.: MIT Press.

Kim, H. (1993). *Qualitative Reasoning about Fluids and Mechanics.* Evanston, Ill.: Institute for Learning Sciences, Northwestern University.

Kuipers, B. (1986). Qualitative simulation. *Artificial Intelligence, 29*(3), 289-338. doi:http://dx.doi.org/10.1016/0004-3702(86)90073-1

Lakoff, G., & Johnson, M. (1980). *Metaphors We Live By.* Chicago: University of Chicago Press.

LaValle, S. (2006). *Planning Algorithms.* Cambridge: Cambridge University Press.

Lenat, D., Prakash, M., & Shepherd, M. (1985). CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *AI Magazine, 6*(4), 65-85.

Lerer, A., Gross, S., & Fergus, R. (2016, March). *Learning Physical Intuition of Block Towers by Example.* Retrieved from preprint arXiv:1603.01312: https://arxiv.org/abs/1603.01312

Marcus, G. (2001). *The Algebraic Mind: Integrating Connectionism and Cognitive Science.* Cambridge, Mass.: MIT Press.

Marr, D. (1982). *Vision.* San Francisco: W.H. Freeman.

Newell, A. (1981). The Knowledge Level. *AI Magazine, 2*(2), 1-20.

Pratt, I., & Schoop, D. (1998). A complete axiom system for polygonal mereotopolgy of the real plane. *Journal of Philosophical Logic, 27*(6), 621-658.

Pratt-Hartmann, ,. I. (2007). First-Order Mereotopology. In J. Aiello, I. Pratt-Hartmann, & J. van Benthem, *Handbook of Spatial Logics.* New York: Springer.

Randell, D., Cui, Z., & Cohn, A. (1992). A spatial logic based on regions and connection. *Proc. 3rd Int. Conf. on Knowledge Representation and Reasoning* (pp. 165-176). Morgan Kaufmann.

Reddy, M. (1979). The Conduit Metaphor. In A. Ortony, *Metaphor and Thought.* Cambridge: Cambridge University Press.

Reece, J., Urry, L., Cain, M., Wasserman, S., Minorsky, P., & Jackson, R. (2011). *Campbell Biology.* (9th, Ed.) Boston: Benjamin Cummings.

Reiter, R. (2001). *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems.* Cambridge, Mass.: MIT Press.

Sandewall, E. (1995). *Features and Fluents: The Representation of Knowledge about Dynamical Systems.* Oxford: Oxford University Press.

Schubert, L. (1990). Monotonic Solution of the Frame Problem in Situation Calculus. In H. Kyburg, R. Loui, & G. Carlson, *Knowledge Representation and Defeasible Reasoning* (pp. 23-67). KLuwer.

Smith, K., Dechter, E., Tenenbaum, J., & Vul, E. (2013). Physical predictions over time. *Proceedings of the 35th Annual Meeting of the Cognitive Science Society.* From http://www.academia.edu/3689251/Physical_predictions_over_time

Stewart, D. (2011). *Dynamics with Inequalities: Impacts and Hard Constraints.* Philadelphia: Society of Industrial and Applied Mathematics.

Weidenbach, C., Dimova, D., Fietzke, A., Kumar, R., Suda, M., & Wischnewski, C. (2009). SPASS Version 3.5. *22 Intl. Conf. on Automated Deduction (CADE), LNCS 5563*, pp. 140-145.